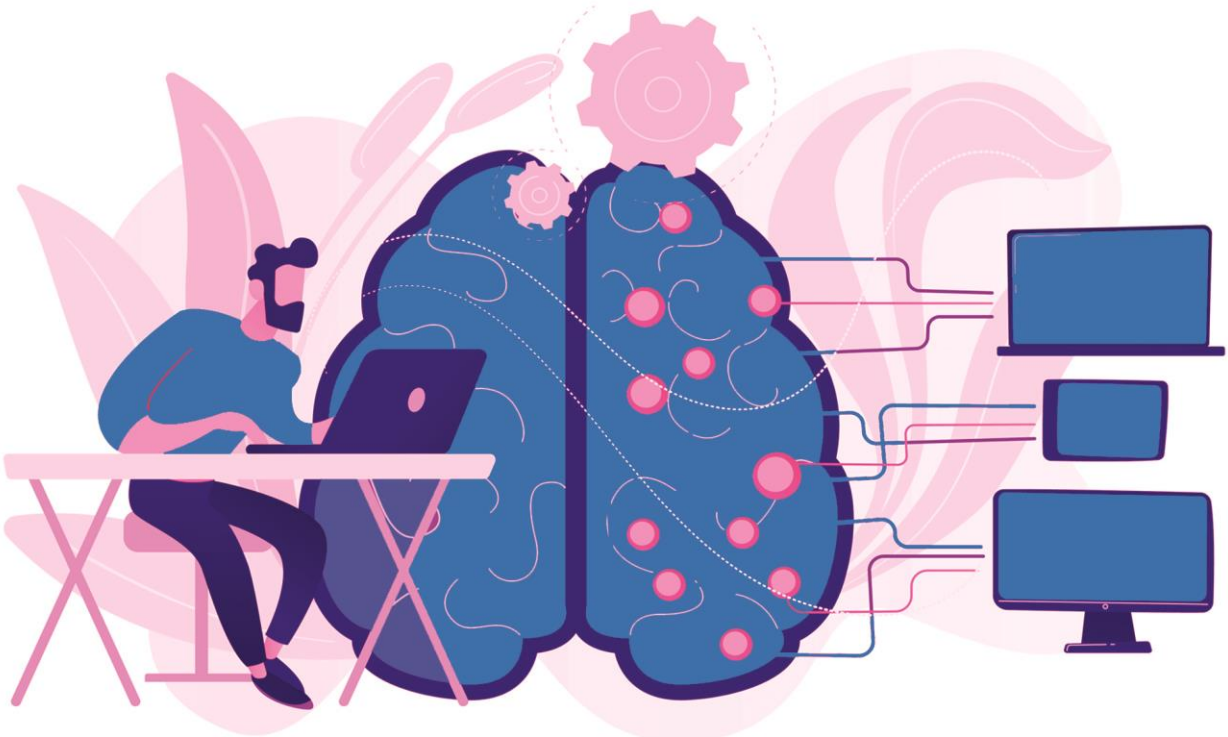
علم البيانات
وتعلم الآلة

45 مشروع علم بيانات مع التعلم الآلي تم حلها وشرحها باستخدام بايثون

ترجمة واعداد : د. علاء طعيمة



بـــمه تعالى

علم البيانات وتعلم الآلة: عن طريق الامثلة

45 مشروع علم البيانات مع التعلم الآلي تم حلها وشرحها باستخدام بايثون

ترجمة واعداد:

د. علاء طعيمة

المقدمة

يعمل التعلم الآلي على إحداث ثورة تدريجيةفي كل مجال من مجالات الذكاء الاصطناعي، مما يجعل تطوير التطبيقات أسهل.

تعد البيانات اليوم أداة ووقودًا للشركات لاكتساب رؤى مهمة وتحسين أدائها. سيطر علم البيانات على كل صناعة تقريبًافي العالم. لا توجد صناعةفي العالم اليوم لا تستخدم البيانات.

في هذا الكتاب، تنقل مشاريع التعلم الآلي باستخدام بايثون كل المعرفة اللازمة لتنفيذ مشاريع التعلم الآليفي مختلف مجالات علم البيانات. كل مشروع من هذه المشاريع فريد من نوعه، مما يساعدك على إتقان الموضوع تدريجيًا.في هذه الكتاب، سوف يأخذك المؤلف من خلال قائمة مشاريع علم البيانات باستخدام التعلم الآلي والتي ستساعدك على تعلم وتنفيذ جميع مفاهيم علم البيانات باستخدام لغة برمجة بايثون.

لقد حاولت قدر المستطاع ان اترجم المشاريع الأكثر طرحاًفي مجال علم البيانات والتعلم الآلي مع الشرح المناسب والكافي، ومع هذا يبقى عملاً بشرياً يحتمل النقص، فاذا كان لديك أي ملاحظات حول هذا الكتاب، فلا تتردد بمراسلتنا عبر بريدنا الالكتروني alaa.taima@qu.edu.iq .

نأمل ان يساعد هذا الكتاب كل من يريد ان يدخلفي مجالات علم البيانات والتعلم الآلي والتعلم العميق وعلم البيانات ومساعدة القارئ العربي على تعلم هذا المجالات. اسأل الله التوفيقفي هذا العمل لأثراء المحتوى العربي الذي يفتقر أشد الافتقار إلى محتوى جيد ورصينفي مجال التعلم الآلي والتعلم العميق وعلم البيانات. ونرجو لك الاستمتاع مع الكتاب ولا تنسونا من صالح الدعاء.

د. علاء طعيمة

كلية علوم الحاسوب وتكنولوجيا المعلومات

جامعة القادسية

العراق

المحتويات

1) التنبؤ بالراتب مع التعلم الآلي Salary Prediction with Machine Learning 13

التنبؤ بالراتب مع التعلم الآلي 13

التنبؤ بالراتب باستخدام بايثون 13

تدريب نموذج التعلم الآلي 14

الملخص 15

2) تجميع بطاقات الائتمان مع التعلم الآلي Credit Card Clustering with Machine Learning 16

تجميع بطاقات الائتمان 16

تجميع بطاقات الائتمان باستخدام بايثون 16

الملخص 20

3) التنبؤ بإيجار المنزل باستخدام التعلم الآلي House Rent Prediction with Machine Learning 21

التنبؤ بإيجار المنزل 21

التنبؤ بإيجار المنزل باستخدام لغة بايثون 21

نموذج التنبؤ بإيجار المنزل 26

الملخص 29

4) مدقق قوة كلمة المرور مع التعلم الآلي Password Strength Checker with Machine Learning 30

كيفية إنشاء مدقق قوة كلمة المرور؟ 30

مدقق قوة كلمة المرور باستخدام بايثون 30

نموذج التنبؤ بقوة كلمة المرور 31

الملخص 32

5) اكتشاف التعليقات غير المرغوب فيها باستخدام التعلم الآلي Spam Comments Detection with Machine Learning 33

اكتشاف التعليقات غير المرغوب فيها 33

اكتشاف التعليقات غير المرغوب فيها باستخدام بايثون 33

تدريب نموذج التصنيف 35

الملخص 35
6) التنبؤ بطلب الطعام عبر الإنترنت باستخدام التعلم الآلي Online Food Order Prediction with Machine Learning 36
التنبؤ بطلب الطعام عبر الإنترنت: دراسة حالة 36
التنبؤ بطلبات الطعام عبر الإنترنت باستخدام لغة بايثون 36
نموذج التنبؤ بطلب الطعام عبر الإنترنت 42
الملخص 44
7) نظام توصية Instagram مع تعلم الآلة Instagram Recommendation System with Machine Learning 45
كيف يعمل نظام توصيات Instagram؟ 45
نظام توصية Instagram باستخدام بايثون 46
الملخص 48
8) التنبؤ بدرجات الطالب مع التعلم الآلي Student Marks Prediction with Machine Learning 49
التنبؤ بدرجات الطالب (دراسة حالة) 49
التنبؤ بدرجات الطالب باستخدام بايثون 49
نموذج التنبؤ بعلامات الطالب 51
الملخص 52
9) كشف الاحتيال في المدفوعات عبر الإنترنت باستخدام التعلم الآلي Online Payments Fraud Detection with Machine Learning 53
كشف الاحتيال في المدفوعات عبر الإنترنت باستخدام التعلم الآلي 53
كشف الاحتيال للمدفوعات عبر الإنترنت باستخدام بايثون 54
نموذج كشف الاحتيال للمدفوعات عبر الإنترنت 56
الملخص 57
10) التنبؤ بإكرامية النادل مع التعلم الآلي Waiter Tips Prediction with Machine Learning 58
اكرامية النادل (دراسة حالة) 58
التنبؤ بإكرامية النادل باستخدام بايثون 58
تحليل اكرامية النادل 59

نموذج التنبؤ بإكرامية النادل 62
الملخص 63
11) تجميع أنواع الموسيقى باستخدام التعلم الآلي Clustering Music Genres with Machine Learning 64
تجميع أنواع الموسيقى (بيان المشكلة) 64
تجميع أنواع الموسيقى باستخدام لغة بايثون 64
التحليل العنقودي للميزات الصوتية 66
الملخص 68
12) التنبؤ بأسعار العملات المشفرة باستخدام التعلم الآلي Cryptocurrency Price Prediction with Machine Learning 69
التنبؤ بأسعار العملات المشفرة باستخدام التعلم الآلي 69
التنبؤ بأسعار العملات المشفرة باستخدام بايثون 69
نموذج التنبؤ بأسعار العملات المشفرة 71
الملخص 72
13) التنبؤ بسعر السهم باستخدام التعلم الآلي Stock Price Prediction using Machine Learning 73
التنبؤ بسعر السهم 73
تحضير البيانات 73
تطبيق تعلم الآلة للتنبؤ بأسعار الأسهم 74
الملخص 75
14) التنبؤ بالنجاة من سرطان الثدي مع التعلم الآلي Breast Cancer Survival Prediction with Machine Learning 76
التنبؤ بالنجاة من سرطان الثدي مع التعلم الآلي 76
التنبؤ بالنجاة من سرطان الثدي باستخدام لغة بايثون 77
نموذج التنبؤ بالنجاة من سرطان الثدي 82
الملخص 83
15) التنبؤ بوفيات Covid-19 باستخدام التعلم الآلي Covid-19 Deaths Prediction with Machine Learning 84
التنبؤ بوفيات كوفيد-19 (دراسة حالة) 84

التنبؤ بوفيات Covid-19 باستخدام بايثون..... 84
تحليل معدل الوفيات Covid-19..... 85
نموذج التنبؤ بوفيات Covid-19 87
الملخص 88
16) التنبؤ بسعر السهم مع LSTM Stock Price Prediction with LSTM ... 89
التنبؤ بسعر السهم مع LSTM 89
تدريب LSTM للتنبؤ بسعر السهم..... 91
الملخص 93
17) نظام توصية المقالات مع التعلم الآلي Article Recommendation System with Machine Learning 94
نظام التوصية بالمقالات 94
نظام التوصية بالمقالات باستخدام بايثون 94
الملخص 96
18) التنبؤ بالمبيعات المستقبلية مع التعلم الآلي Future Sales Prediction with Machine Learning..... 97
التنبؤ بالمبيعات المستقبلية (دراسة حالة)..... 97
التنبؤ بالمبيعات المستقبلية باستخدام بايثون..... 97
نموذج التنبؤ بالمبيعات المستقبلية 100
الملخص 100
19) التنبؤ بسعر سهم Netflix مع التعلم الآلي Netflix Stock Price Prediction with Machine Learning 101
التنبؤ بسعر سهم Netflix مع التعلم الآلي 101
تدريب LSTM على التنبؤ بسعر سهم Netflix..... 102
الملخص 105
20) اكتشاف الاجهاد باستخدام التعلم الآلي Stress Detection with Machine Learning..... 106
اكتشاف الاجهاد باستخدام التعلم الآلي 106
اكتشاف الإجهاد باستخدام بايثون 106
نموذج اكتشاف الإجهاد 108

الملخص 110

21) التنبؤ بالطلب على المنتج باستخدام التعلم الآلي Product Demand Prediction with Machine Learning 111

التنبؤ بالطلب على المنتج (دراسة حالة) 111

التنبؤ بالطلب على المنتج باستخدام بايثون 111

نموذج التنبؤ بالطلب على المنتج 113

الملخص 114

22) التنبؤ بأسعار الكهرباء مع التعلم الآلي Electricity Price Prediction with Machine Learning 115

التنبؤ بأسعار الكهرباء (دراسة حالة) 115

التنبؤ بأسعار الكهرباء باستخدام لغة بايثون 116

نموذج التنبؤ بأسعار الكهرباء 118

الملخص 119

23) اكتشاف اللغة باستخدام التعلم الآلي Language Detection with Machine Learning 120

اكتشاف اللغة 120

اكتشاف اللغة باستخدام بايثون 121

نموذج اكتشاف اللغة 122

الملخص 122

24) التنبؤ بسعر سهم Tata Motors مع التعلم الآلي Tata Motors Stock Price Prediction with Machine Learning 124

التنبؤ بسعر سهم Tata Motors 124

التنبؤ بأسعار أسهم شركة Tata Motors باستخدام لغة بايثون 124

الملخص 126

25) التنبؤ بأقساط التأمين الصحي باستخدام التعلم الآلي Health Insurance Premium Prediction with Machine Learning 127

التنبؤ بأقساط التأمين الصحي 127

التنبؤ بأقساط التأمين الصحي باستخدام بايثون 127

نموذج التنبؤ بأقساط التأمين الصحي 129

الملخص 130
26) التنبؤ بعدد الطلبات باستخدام التعلم الآلي Number of Orders Prediction with Machine Learning 131
التنبؤ بعدد الطلبات 131
التنبؤ بعدد الطلبات باستخدام بايثون 132
نموذج التنبؤ بعدد الطلبات 135
الملخص 136
27) التنبؤ بسعر سهم Apple مع التعلم الآلي Apple Stock Price Prediction with Machine Learning 137
التنبؤ بسعر سهم Apple..... 137
التنبؤ بسعر سهم Apple باستخدام بايثون..... 137
الملخص 139
28) تصنيف الأخبار مع تعلم الآلة News Classification with Machine Learning 140
تصنيف الأخبار 140
تصنيف الأخبار باستخدام بايثون 140
نموذج تصنيف الأخبار 141
الملخص 142
29) التنبؤ بأسعار السيارة مع التعلم الآلي Car Price Prediction with Machine Learning..... 143
التنبؤ بأسعار السيارة مع التعلم الآلي..... 143
نموذج التنبؤ بسعر السيارة باستخدام لغة بايثون 143
تدريب نموذج التنبؤ بسعر السيارة 148
الملخص..... 148
30) اكتشاف البريد العشوائي باستخدام التعلم الآلي Spam Detection with Machine Learning..... 149
اكتشاف البريد العشوائي..... 149
اكتشاف البريد العشوائي باستخدام بايثون..... 149
الملخص..... 150

31) التنبؤ بدرجات الطلاب باستخدام التعلم الآلي Student Grades Prediction with Machine Learning 151
التنبؤ بدرجات الطالب 151
التنبؤ بدرجات الطالب باستخدام لغة بايثون 151
الملخص 152
32) التنبؤ بسعر سهم Tesla مع التعلم الآلي Tesla Stock Price Prediction with Machine Learning 153
التنبؤ بسعر سهم Tesla مع التعلم الآلي 153
التنبؤ بسعر سهم Tesla باستخدام لغة بايثون 153
الملخص 155
33) التنبؤ بمعدل النقر للظهور من خلال التعلم الآلي Click-Through Rate Prediction with Machine Learning 156
نموذج التنبؤ بمعدل النقر مع التعلم الآلي 156
نموذج التنبؤ بمعدل النقر باستخدام بايثون 156
نموذج الانحدار اللوجستي: 158
34) اكتشاف السخرية مع التعلم الآلي Sarcasm Detection with Machine Learning 160
اكتشاف السخرية مع التعلم الآلي 160
اكتشاف السخرية باستخدام بايثون 160
الملخص 162
35) التنبؤ بمتابعي وسائل التواصل الاجتماعي باستخدام التعلم الآلي Social Media Followers Prediction with Machine Learning 163
التنبؤ بمتابعي وسائل التواصل الاجتماعي 163
التنبؤ بمتابعي الوسائط الاجتماعية باستخدام بايثون 164
الملخص 166
36) اكتشاف خطاب الكراهية باستخدام التعلم الآلي Hate Speech Detection with Machine Learning 167
اكتشاف خطاب الكراهية باستخدام التعلم الآلي 167
اكتشاف خطاب الكراهية باستخدام بايثون 167
الملخص 170

37) التنبؤ بسعر Dogecoin مع التعلم الآلي Dogecoin Price Prediction with Machine Learning 171
التنبؤ بسعر Dogecoin 171
التنبؤ بسعر Dogecoin باستخدام بايثون..... 171
الملخص 173
38) تصنيف إعلانات وسائل التواصل الاجتماعي مع التعلم الآلي Social Media Ads Classification with Machine Learning 174
تصنيف إعلانات وسائل التواصل الاجتماعي 174
تصنيف إعلانات وسائل التواصل الاجتماعي باستخدام بايثون 174
تدريب نموذج تصنيف إعلانات وسائل التواصل الاجتماعية..... 176
الملخص 177
39) اكتشاف الأخبار الكاذبة باستخدام التعلم الآلي Fake News Detection with Machine Learning 178
اكتشاف الأخبار الكاذبة 178
اكتشاف الأخبار الكاذبة باستخدام بايثون..... 178
الملخص 180
40) التنبؤ بالمبيعات باستخدام التعلم الآلي Sales Prediction with Machine Learning..... 181
ما هو التنبؤ بالمبيعات؟..... 181
التنبؤ بالمبيعات باستخدام بايثون..... 181
الملخص..... 183
41) تصنيف أسعار الأجهزة المحمولة مع التعلم الآلي Mobile Price Classification with Machine Learning 184
تصنيف أسعار الأجهزة المحمولة مع التعلم الآلي 184
تصنيف أسعار الأجهزة المحمولة باستخدام لغة بايثون..... 184
تحضير البيانات..... 186
نموذج تصنيف سعر الهاتف المحمول 186
43) التنبؤ بسعر صرف العملات مع التعلم الآلي Currency Exchange Rate Prediction with Machine Learning 188
التنبؤ بسعر صرف العملات 188

التنبؤ بسعر صرف العملات باستخدام بايثون 188
الملخص 191
44) التنبؤ بالربح مع التعلم الآلي Profit Prediction with Machine Learning 192
التنبؤ بالربح مع التعلم الآلي 192
التنبؤ بالربح باستخدام بايثون 192
الملخص 194
45) تحليل جودة المياه باستخدام التعلم الآلي Water Quality Analysis using Machine Learning 195
تحليل جودة المياه..... 195
تحليل جودة المياه باستخدام لغة بايثون..... 195
نموذج التنبؤ بجودة المياه باستخدام لغة بايثون 201
الملخص..... 202
46) التنبؤ بمبيعات ألعاب الفيديو باستخدام التعلم الآلي Video Game Sales Prediction with Machine Learning 203
نموذج التنبؤ بمبيعات ألعاب الفيديو باستخدام بايثون 203
نموذج التنبؤ بمبيعات ألعاب الفيديو التدريبية 205
الملخص 205

1) التنبؤ بالراتب مع التعلم الآلي Salary Prediction with Machine Learning

يختلف الراتب (Salary) باختلاف الملف الوظيفي للشخص. لكن بشكل عام، فإن خبرة العمل هي التي تحدد الراتب. يعد التنبؤ بالراتب (Salary prediction) مشكلة شائعة بين مجتمع علوم البيانات للمبتدئين تمامًا. لذلك، إذا كنت مبتدئًافي علوم البيانات، فيجب أن تعمل على حل هذه المشكلة لفهم التعلم الآلي (Machine Learning).في هذه المقالة، سوف نخوضفي مهمة التنبؤ بالراتب باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بالراتب مع التعلم الآلي

للتنبؤ بالراتب، نحتاج إلى إيجاد علاقاتفي البيانات حول كيفية تحديد الراتب. لهذه المهمة، نحتاج إلى مجموعة بيانات تعتمد على الرواتب. لقد وجدت مجموعة بيانات تحتوي على بيانات حول كيفية تأثير الخبرة الوظيفية (job experience) على الراتب.

تحتوي مجموعة البيانات على عمودين فقط:

1. الخبرة الوظيفية (job experience).
2. الراتب (salary).

يمكنك تنزيل مجموعة البيانات من هنا.

في القسم أدناه، سأقدم كيفية استخدام التعلم الآلي للتنبؤ بالراتب بناءً على الخبرة الوظيفية.

التنبؤ بالراتب باستخدام بايثون

لنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("Salary_Data.csv")
print(data.head())
```

```
   YearsExperience   Salary
0              1.1  39343.0
1              1.3  46205.0
2              1.5  37731.0
3              2.0  43525.0
4              2.2  39891.0
```

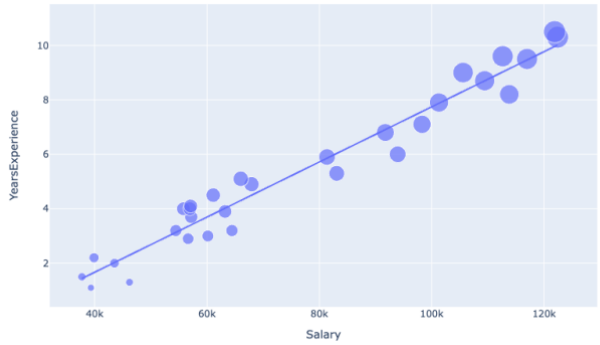
دعنا نتحقق مما إذا كانت مجموعة البيانات تحتوي على أي قيم فارغة (null values) أم لا:

```
print(data.isnull().sum())
```

```
YearsExperience    0
Salary             0
dtype: int64
```

لا تحتوي مجموعة البيانات على أي قيم خالية. دعونا نلقي نظرة على العلاقة بين الراتب والخبرة الوظيفية للأفراد:

```
figure = px.scatter(data_frame = data ,
                    x="Salary,"
                    y="YearsExperience ,"
                    size="YearsExperience ,"
                    trendline="ols")
figure.show()
```



هناك علاقة خطية مثالية بين الراتب والخبرة الوظيفية للناس. هذا يعني أن المزيد من الخبرة الوظيفية يؤدي إلى راتب أعلى.

تدريب نموذج التعلم الآلي

نظرًا لأن هذه مشكلة تحليل الانحدار (regression analysis)، فسوف نقوم بتدريب نموذج الانحدار (regression model) للتنبؤ بالراتب باستخدام التعلم الآلي. إليك كيفية تقسيم البيانات إلى مجموعات تدريب واختبار قبل تدريب النموذج:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

x = np.asanyarray(data[["YearsExperience"]])
y = np.asanyarray(data[["Salary"]])
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                test_size=0.2,
```

```
random_state=42)
```

إليك الآن كيف يمكننا تدريب نموذج التعلم الآلي:

```
model = LinearRegression()
model.fit(xtrain, ytrain)
```

الآن دعنا نتوقع راتب شخص ما باستخدام نموذج التعلم الآلي المدرب:

```
a = float(input("Years of Experience : "))
features = np.array([[a]])
print("Predicted Salary = ", model.predict(features))
```

```
Years of Experience : 2
Predicted Salary =  [[44169.21365784]]
```

إذن هذه هي الطريقة التي يمكنك بها حل مشكلة التنبؤ بالراتب كمبتدئفي علوم البيانات.

الملخص

يعد التنبؤ بالراتب مشكلة شائعة بين مجتمع علوم البيانات للمبتدئين تمامًا. من خلال تحليل الانحدار هذا، وجدنا علاقة خطية مثالية بين الراتب والخبرة الوظيفية للأفراد. هذا يعني أن المزيد من الخبرة الوظيفية يؤدي إلى راتب أعلى. آمل أن تكون قد أحببت هذه المقالة حول مهمة التنبؤ بالراتب باستخدام التعلم الآلي باستخدام بايثون.

2) تجميع بطاقات الائتمان مع التعلم الآلي Credit Card Clustering with Machine Learning

تجميع بطاقات الائتمان (Credit card clustering) هو مهمة تجميع حاملي بطاقات الائتمان بناءً على عادات الشراء وحدود الائتمان والعديد من العوامل المالية الأخرى. إذا كنت تريد معرفة كيفية استخدام تحليل المجموعات لتجميع حاملي بطاقات الائتمان، فهذه المقالة مناسبة لك. في هذه المقالة، سوف آخذك خلال مهمة تجميع بطاقات الائتمان باستخدام التعلم الآلي باستخدام بايثون.

تجميع بطاقات الائتمان

يعني تجميع بطاقات الائتمان تجميع حاملي بطاقات الائتمان بناءً على عادات الشراء (buying habits) وحدود الائتمان (credit limits) والعديد من العوامل المالية الأخرى. يُعرف أيضًا باسم تجزئة بطاقة الائتمان (credit card segmentation). يساعد تحليل المجموعات هذا الشركاتفي العثور على عملائها المحتملين والعديد من استراتيجيات التسويق الأخرى.

بالنسبة لمهمة تجميع بطاقات الائتمان مع التعلم الآلي، نحتاج إلى مجموعة بيانات تستند إلى سجل الشراء لحاملي بطاقات الائتمان. لقد وجدت مجموعة بيانات مثالية لهذه المهمة تحتوي على جميع الميزات الضرورية التي تكفي لفهم تحليل مجموعة بطاقات الائتمان. يمكنك تنزيل مجموعة البيانات من هنا.

في القسم أدناه، سوف آخذك خلال مهمة تحليل مجموعات بطاقات الائتمان باستخدام التعلم الآلي باستخدام لغة برمجة بايثون.

تجميع بطاقات الائتمان باستخدام بايثون

لنبدأ مهمة تحليل مجموعة بطاقات الائتمان عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
from sklearn import cluster

data = pd.read_csv("CC GENERAL.csv")
print(data.head())
```

```
  CUST_ID      BALANCE  BALANCE_FREQUENCY  PURCHASES  ONEOFF_PURCHASES  \
0  C10001    40.900749           0.818182      95.40              0.00
1  C10002  3202.467416           0.909091       0.00              0.00
2  C10003  2495.148862           1.000000     773.17            773.17
3  C10004  1666.670542           0.636364    1499.00           1499.00
4  C10005   817.714335           1.000000      16.00             16.00

   INSTALLMENTS_PURCHASES  CASH_ADVANCE  PURCHASES_FREQUENCY  \
0                    95.4      0.000000             0.166667
1                     0.0   6442.945483             0.000000
2                     0.0      0.000000             1.000000
3                     0.0    205.788017             0.083333
4                     0.0      0.000000             0.083333

   ONEOFF_PURCHASES_FREQUENCY  PURCHASES_INSTALLMENTS_FREQUENCY  \
0                    0.000000                          0.083333
1                    0.000000                          0.000000
2                    1.000000                          0.000000
3                    0.083333                          0.000000
4                    0.083333                          0.000000

   CASH_ADVANCE_FREQUENCY  CASH_ADVANCE_TRX  PURCHASES_TRX  CREDIT_LIMIT  \
0                0.000000                 0              2        1000.0
1                0.250000                 4              0        7000.0
2                0.000000                 0             12        7500.0
3                0.083333                 1              1        7500.0
4                0.000000                 0              1        1200.0

      PAYMENTS  MINIMUM_PAYMENTS  PRC_FULL_PAYMENT  TENURE
0   201.802084        139.509787          0.000000      12
1  4103.032597       1072.340217          0.222222      12
2   622.066742        627.284787          0.000000      12
3     0.000000               NaN          0.000000      12
4   678.334763        244.791237          0.000000      12
```

قبل المضي قدمًا، دعنا نتحقق مما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
CUST_ID                           0
BALANCE                           0
BALANCE_FREQUENCY                 0
PURCHASES                         0
ONEOFF_PURCHASES                  0
INSTALLMENTS_PURCHASES            0
CASH_ADVANCE                      0
PURCHASES_FREQUENCY               0
ONEOFF_PURCHASES_FREQUENCY        0
PURCHASES_INSTALLMENTS_FREQUENCY  0
CASH_ADVANCE_FREQUENCY            0
CASH_ADVANCE_TRX                  0
PURCHASES_TRX                     0
CREDIT_LIMIT                      1
PAYMENTS                          0
MINIMUM_PAYMENTS                313
PRC_FULL_PAYMENT                  0
TENURE                            0
dtype: int64
```

تحتوي مجموعة البيانات على بعض القيم الخاليةفي عمود الحد الأدنى للمدفوعات (minimum payments column). سوف أسقط الصفوف ذات القيم الفارغة وسأنتقل إلى أبعد من ذلك:

```
data = data.dropna()
```

هناك ثلاث ميزاتفي مجموعة البيانات تعتبر ذات قيمة كبيرة لمهمة تقسيم بطاقة الائتمان:

1. BALANCE: الرصيد المتبقيفي حسابات عملاء بطاقات الائتمان.
2. PURCHASES: مقدار المشتريات التي تمت من حسابات عملاء بطاقات الائتمان.
3. CREDIT_LIMIT: حد بطاقة الائتمان.

هذه الميزات الثلاث كافية لتجميع حاملي بطاقات الائتمان حيث يخبروننا عن تاريخ الشراء (buying history)، والرصيد المصرفي (bank balance)، والحد الائتماني (credit limit) لحاملي بطاقات الائتمان. فلنستخدم هذه الميزات لإنشاء مجموعات من مجموعة البيانات:

```
clustering_data = data[["BALANCE", "PURCHASES",
"CREDIT_LIMIT"]]
from sklearn.preprocessing import MinMaxScaler
for i in clustering_data.columns:
  MinMaxScaler(i)

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=5)
clusters = kmeans.fit_predict(clustering_data)
data["CREDIT_CARD_SEGMENTS"] = clusters
```

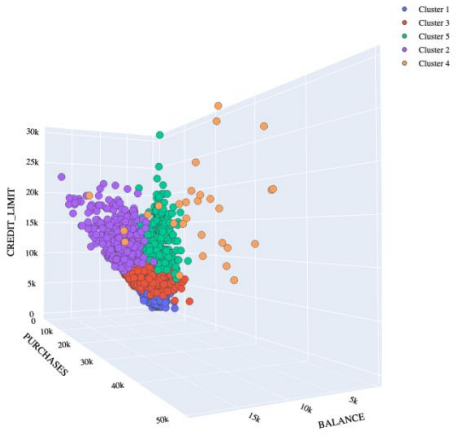
لقد أضفت عمودًا جديدًا باسم "CREDIT_CARD_SEGMENTS". يحتوي على تسميات (labels) حول مجموعة عملاء بطاقات الائتمان. تتراوح المجموعات المكونة من 0 إلى 4. للتبسيط، سأقوم بتحويل أسماء هذه المجموعات:

```
data["CREDIT_CARD_SEGMENTS"] =
data["CREDIT_CARD_SEGMENTS"].map({0: "Cluster 1", 1:
"  Cluster 2", 2: "Cluster 3", 3: "Cluster 4", 4: "Cluster 5"})}
print(data["CREDIT_CARD_SEGMENTS"].head(10))
```

```
0    Cluster 1
1    Cluster 3
2    Cluster 3
4    Cluster 1
5    Cluster 1
6    Cluster 5
7    Cluster 1
8    Cluster 3
9    Cluster 5
10   Cluster 1
Name: CREDIT_CARD_SEGMENTS, dtype: object
```

الآن دعونا نرسم مجموعات بطاقات الائتمان التي وجدناها من تحليل المجموعة لدينا:

```
import plotly.graph_objects as go
PLOT = go.Figure()
for i in list(data["CREDIT_CARD_SEGMENTS"].unique()):


  PLOT.add_trace(go.Scatter3d(x =
data[data["CREDIT_CARD_SEGMENTS"]== i]['BALANCE,['
             y = data[data["CREDIT_CARD_SEGMENTS"] ==
i]['PURCHASES'],
             z = data[data["CREDIT_CARD_SEGMENTS"] ==
i]['CREDIT_LIMIT']          ,
             mode = 'markers',marker_size = 6,
marker_line_width = 1,
             name = str(i)((
PLOT.update_traces(hovertemplate='BALANCE: %{x} <br>PURCHASES
%{y} <br>DCREDIT_LIMIT: %{z}')


PLOT.update_layout(width = 800, height = 800, autosize = True,
showlegend = True,
        scene = dict(xaxis=dict(title = 'BALANCE',
titlefont_color = 'black,('
             yaxis=dict(title = 'PURCHASES', titlefont_color =
'black'),
             zaxis=dict(title = 'CREDIT_LIMIT',
titlefont_color = 'black'),(
                   font = dict(family = "Gilroy", color  =
'black', size = 12))
```



هذه هي الطريقة التي يمكنك بها إجراء تجزئة لبطاقات الائتمان باستخدام التعلم الآلي باستخدام بايثون.

الملخص

يعني تحليل مجموعة بطاقات الائتمان تجميع حاملي بطاقات الائتمان بناءً على عادات الشراء وحدود الائتمان والعديد من العوامل المالية الأخرى. يساعد تحليل المجموعات هذا الشركات في العثور على عملائها المحتملين والعديد من استراتيجيات التسويق الأخرى. آمل أن تكون قد أحببت هذه المقالة حول تجزئة بطاقة الائتمان باستخدام التعلم الآلي باستخدام بايثون.

3) التنبؤ بإيجار المنزل باستخدام التعلم الآلي House Rent Prediction with Machine Learning

يعتمد إيجار المنزل (House Rent) على العديد من العوامل. مع البيانات المناسبة وتقنيات التعلم الآلي، تجد العديد من المنصات العقارية خيارات الإسكان وفقًا لميزانية العميل. لذلك، إذا كنت تريد معرفة كيفية استخدام التعلم الآلي للتنبؤ بإيجار منزل، فهذه المقالة مناسبة لك.في هذه المقالة، سوف نخوضفي مهمة التنبؤ بإيجار المنزل باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بإيجار المنزل

يعتمد إيجار عقار سكني على العديد من العوامل مثل:

1. عدد غرف النوم والصالة والمطبخ (number of bedrooms, hall, and kitchen).
2. حجم العقار (size of the property).
3. أرضية المنزل (the floor of the house).
4. نوع المنطقة (area type).
5. محلة المنطقة (area locality).
6. المدينة (City).
7. حالة تأثيث المنزل (furnishing status of the house).

لبناء نظام التنبؤ بإيجار المنزل، نحتاج إلى بيانات تستند إلى العوامل التي تؤثر على إيجار عقار سكني. لقد عثرت على مجموعة بيانات من Kaggle تتضمن جميع الميزات التي نحتاجها. يمكنك تنزيل مجموعة البيانات من هنا.

التنبؤ بإيجار المنزل باستخدام لغة بايثون

سأبدأ مهمة التنبؤ بإيجار المنزل عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("House_Rent_Dataset
```

```
    Posted On  BHK   Rent  Size            Floor    Area Type  \
0  2022-05-18    2  10000  1100  Ground out of 2   Super Area
1  2022-05-13    2  20000   800       1 out of 3   Super Area
2  2022-05-16    2  17000  1000       1 out of 3   Super Area
3  2022-07-04    2  10000   800       1 out of 2   Super Area
4  2022-05-09    2   7500   850       1 out of 2  Carpet Area

             Area Locality     City Furnishing Status  Tenant Preferred  \
0                   Bandel  Kolkata       Unfurnished  Bachelors/Family
1  Phool Bagan, Kankurgachi  Kolkata    Semi-Furnished  Bachelors/Family
2   Salt Lake City Sector 2  Kolkata    Semi-Furnished  Bachelors/Family
3              Dumdum Park  Kolkata       Unfurnished  Bachelors/Family
4            South Dum Dum  Kolkata       Unfurnished         Bachelors

   Bathroom Point of Contact
0         2    Contact Owner
1         1    Contact Owner
2         1    Contact Owner
3         1    Contact Owner
4         1    Contact Owner
```

قبل المضي قدمًا، دعنا نتحقق مما إذا كانت البيانات تحتوي على قيم خالية أم لا:

```
print(data.isnull().sum())
```

```
Posted On           0
BHK                 0
Rent                0
Size                0
Floor               0
Area Type           0
Area Locality       0
City                0
Furnishing Status   0
Tenant Preferred    0
Bathroom            0
Point of Contact    0
dtype: int64
```

دعونا نلقي نظرة على الإحصائيات الوصفية للبيانات:

```
print(data.describe())
```

```
               BHK          Rent          Size     Bathroom
count  4746.000000  4.746000e+03   4746.000000  4746.000000
mean      2.083860  3.499345e+04    967.490729     1.965866
std       0.832256  7.810641e+04    634.202328     0.884532
min       1.000000  1.200000e+03     10.000000     1.000000
25%       2.000000  1.000000e+04    550.000000     1.000000
50%       2.000000  1.600000e+04    850.000000     2.000000
75%       3.000000  3.300000e+04   1200.000000     2.000000
max       6.000000  3.500000e+06   8000.000000    10.000000
```

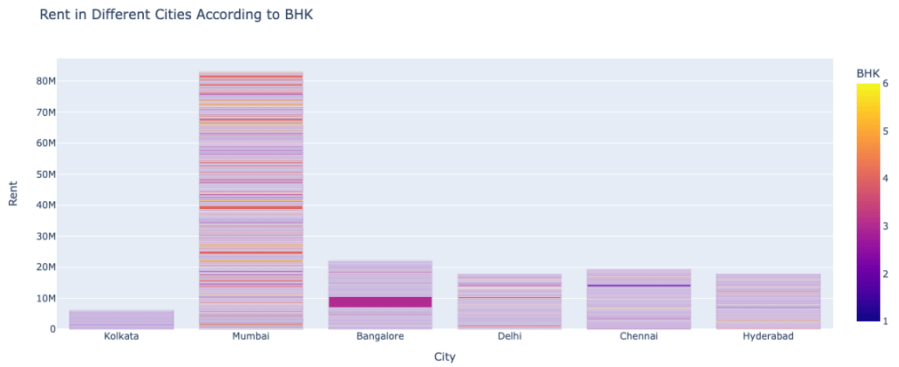
دعنا الآن نلقي نظرة على متوسط (mean) الإيجار ووسيطه (median) وأعلى (highest) وأدنى (lowest) إيجار للمنازل:

```
print(f"Mean Rent: {data.Rent.mean()}")
print(f"Median Rent: {data.Rent.median()}")
print(f"Highest Rent: {data.Rent.max()}")
print(f"Lowest Rent: {data.Rent.min()}")
```

```
Mean Rent: 34993.45132743363
Median Rent: 16000.0
Highest Rent: 3500000
Lowest Rent: 1200
```

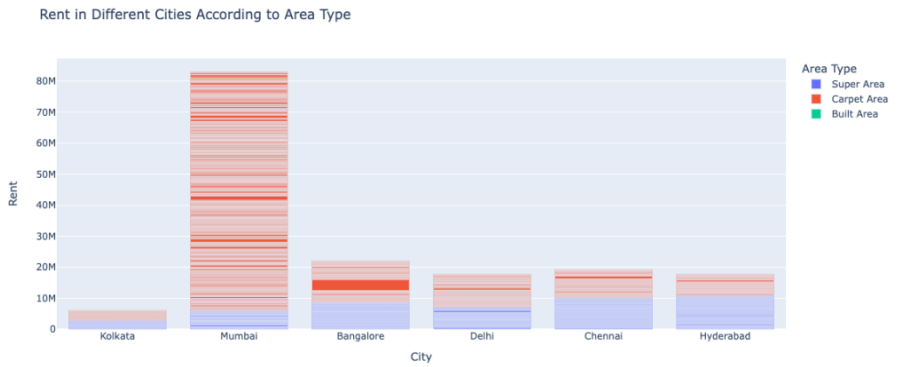
دعونا الآن نلقي نظرة على إيجار المنازلفي مدن مختلفة حسب عدد غرف النوم والصالات والمطابخ:

```
figure = px.bar(data, x=data["City"],
       y = data["Rent"],
       color = data["BHK"],
     title="Rent in Different Cities According to BHK")
figure.show()
```



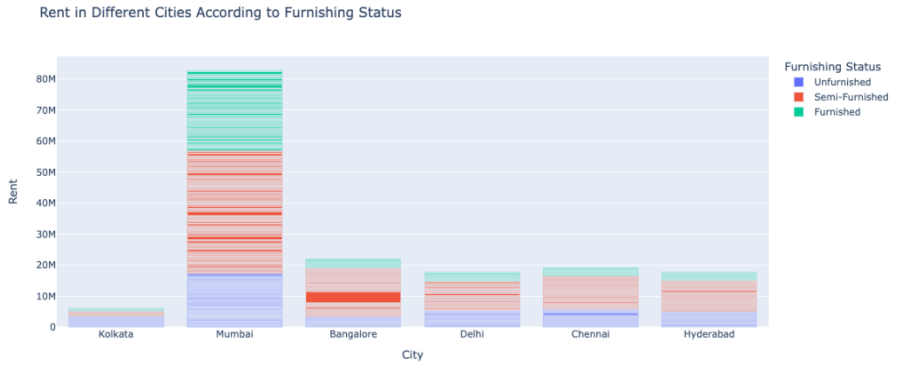
دعونا الآن نلقي نظرة على إيجار المنازلفي مدن مختلفة وفقًا لنوع المنطقة:

```
figure = px.bar(data, x=data["City"],
       y = data["Rent"],
       color = data["Area Type"],
     title="Rent in Different Cities According to Area Type")
figure.show()
```



دعونا الآن نلقي نظرة على إيجار المنازلفي مدن مختلفة وفقًا لحالة تأثيث المنزل:

```
figure = px.bar(data, x=data["City”],
       y = data["Rent"],
       color = data["Furnishing Status"],
     title="Rent in Different Cities According to Furnishing
Status”)
figure.show()
```

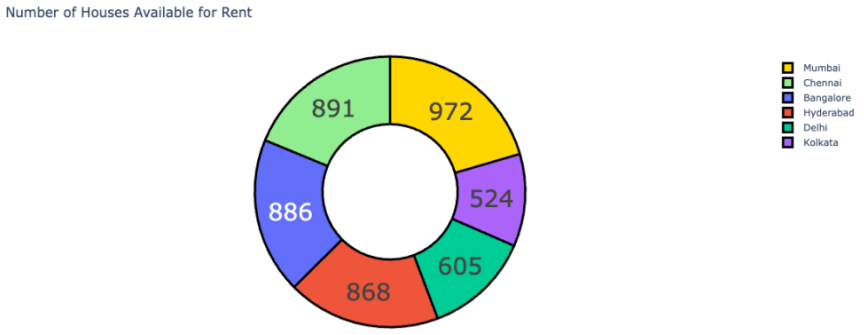


دعونا الآن نلقي نظرة على إيجار المنازلفي مدن مختلفة حسب حجم المنزل:

```
figure = px.bar(data, x=data["City”],
       y = data["Rent"],
       color = data["Size"],
     title="Rent in Different Cities According to Size”)
figure.show()
```

دعونا الآن نلقي نظرة على عدد المنازل المتاحة للإيجارفي مدن مختلفة وفقًا لمجموعة البيانات:

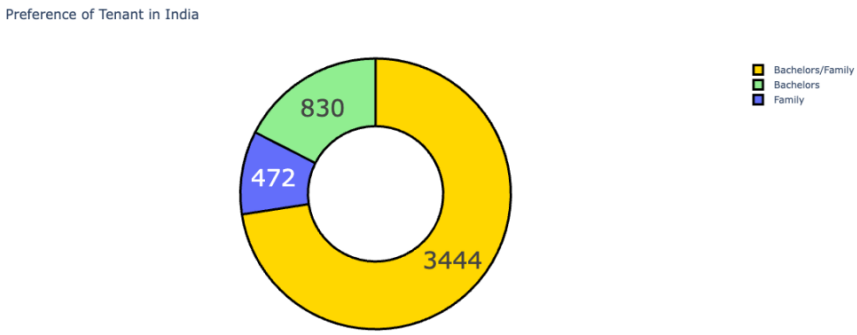
```
cities = data["City"].value_counts()
label = cities.index
counts = cities.values
colors = ['gold','lightgreen']

fig = go.Figure(data=[go.Pie(labels=label, values=counts,
hole=0.5)])
fig.update_layout(title_text='Number of Houses Available for
Rent')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3)))
fig.show()
```



دعونا الآن نلقي نظرة على عدد المنازل المتاحة لأنواع مختلفة من المستأجرين (tenants):

```
#Preference of Tenant
tenant = data["Tenant Preferred"].value_counts()
label = tenant.index
```

```
counts = tenant.values
colors = ['gold','lightgreen']

fig = go.Figure(data=[go.Pie(labels=label, values=counts,
hole=0.5)])
fig.update_layout(title_text='Preference of Tenant in India')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
       marker=dict(colors=colors, line=dict(color='black',
width=3)))
fig.show()
```



نموذج التنبؤ بإيجار المنزل

الآن سأحول جميع الميزات الفئوية (categorical features) إلى ميزات عددية نحتاجها لتدريب نموذج التنبؤ بإيجار المنزل:

```
data["Area Type"] = data["Area Type"].map({"Super Area": 1,
"                 Carpet Area": 2,
"                 Built Area": 3})
data["City"] = data["City"].map({"Mumbai": 4000, "Chennai":
6000,
"             Bangalore": 5600, "Hyderabad": 5000,
"             Delhi": 1100, "Kolkata": 7000})
data["Furnishing Status"] = data["Furnishing
Status"].map({"Unfurnished": 0,
"                      Semi-Furnished": 1,
"                      Furnished": 2})
data["Tenant Preferred"] = data["Tenant
Preferred"].map({"Bachelors/Family": 2,
"                     Bachelors": 1,
"                     Family": 3})
```

```
    Posted On  BHK   Rent  Size            Floor  Area Type  \
0  2022-05-18    2  10000  1100  Ground out of 2          1
1  2022-05-13    2  20000   800       1 out of 3          1
2  2022-05-16    2  17000  1000       1 out of 3          1
3  2022-07-04    2  10000   800       1 out of 2          1
4  2022-05-09    2   7500   850       1 out of 2          2

              Area Locality  City  Furnishing Status  Tenant Preferred  \
0                   Bandel  7000                  0                 2
1  Phool Bagan, Kankurgachi  7000                  1                 2
2   Salt Lake City Sector 2  7000                  1                 2
3              Dumdum Park  7000                  0                 2
4            South Dum Dum  7000                  0                 1

   Bathroom Point of Contact
0         2    Contact Owner
1         1    Contact Owner
2         1    Contact Owner
3         1    Contact Owner
4         1    Contact Owner
```

الآن سأقسم البيانات إلى مجموعات تدريب واختبار:

```
##splitting data
from sklearn.model_selection import train_test_split
x = np.array(data[["BHK", "Size", "Area Type", "City",
                   "Furnishing Status", "Tenant Preferred",
                   "Bathroom"]])
y = np.array(data[["Rent"]])

xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                test_size=0.10,
                                                random_state=42)
```

دعنا الآن ندرب نموذج التنبؤ بإيجار المنزل باستخدام نموذج الشبكة العصبية LSTM:

```
from keras.models import Sequential
from keras.layers import Dense, LSTM
model = Sequential()
model.add(LSTM(128, return_sequences=True,
      input_shape= (xtrain.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 7, 128)            66560

 lstm_1 (LSTM)               (None, 64)                49408

 dense (Dense)               (None, 25)                1625

 dense_1 (Dense)             (None, 1)                 26

=================================================================
Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0
_________________________________________________________________
```

```
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(xtrain, ytrain, batch_size=1, epochs=21)
```

```
Epoch 1/21
4271/4271 [==============================] - 35s 7ms/step - loss: 7038080512.0000
Epoch 2/21
4271/4271 [==============================] - 31s 7ms/step - loss: 6481502720.0000
Epoch 3/21
4271/4271 [==============================] - 31s 7ms/step - loss: 6180754944.0000
Epoch 4/21
4271/4271 [==============================] - 31s 7ms/step - loss: 5968361472.0000
Epoch 5/21
4271/4271 [==============================] - 30s 7ms/step - loss: 5770649088.0000
Epoch 6/21
4271/4271 [==============================] - 29s 7ms/step - loss: 5618835968.0000
Epoch 7/21
4271/4271 [==============================] - 30s 7ms/step - loss: 5440893952.0000
Epoch 8/21
4271/4271 [==============================] - 29s 7ms/step - loss: 5341533696.0000
Epoch 9/21
4271/4271 [==============================] - 30s 7ms/step - loss: 5182846976.0000
Epoch 10/21
4271/4271 [==============================] - 31s 7ms/step - loss: 5106288128.0000
Epoch 11/21
4271/4271 [==============================] - 30s 7ms/step - loss: 5076118528.0000
Epoch 12/21
4271/4271 [==============================] - 30s 7ms/step - loss: 5001080320.0000
Epoch 13/21
4271/4271 [==============================] - 31s 7ms/step - loss: 4941253120.0000
Epoch 14/21
4271/4271 [==============================] - 33s 8ms/step - loss: 4904356864.0000
Epoch 15/21
4271/4271 [==============================] - 29s 7ms/step - loss: 4854262784.0000
Epoch 16/21
4271/4271 [==============================] - 30s 7ms/step - loss: 4855796736.0000
Epoch 17/21
4271/4271 [==============================] - 36s 8ms/step - loss: 4764052480.0000
Epoch 18/21
4271/4271 [==============================] - 30s 7ms/step - loss: 4709226496.0000
Epoch 19/21
4271/4271 [==============================] - 31s 7ms/step - loss: 4702300160.0000
Epoch 20/21
4271/4271 [==============================] - 31s 7ms/step - loss: 4670900736.0000
Epoch 21/21
4271/4271 [==============================] - 31s 7ms/step - loss: 4755582976.0000
<keras.callbacks.History at 0x7fd1deb6c9d0>
```

الآن إليك كيفية التنبؤ بإيجار عقار سكني باستخدام النموذج المدرب:

```
print("Enter House Details to Predict Rent")
a = int(input("Number of BHK: "))
b = int(input("Size of the House: "))
c = int(input("Area Type (Super Area = 1, Carpet Area = 2,
Built Area = 3): "))
d = int(input("Pin Code of the City: "))
e = int(input("Furnishing Status of the House (Unfurnished =
0, Semi-Furnished = 1, Furnished = 2): "))
f = int(input("Tenant Type (Bachelors = 1, Bachelors/Family =
2, Only Family = 3): "))
g = int(input("Number of bathrooms: "))
features = np.array([[a, b, c, d, e, f, g]])
print("Predicted House Price = ", model.predict(features))
```

```
Enter House Details to Predict Rent
Number of BHK: 3
Size of the House: 1100
Area Type (Super Area = 1, Carpet Area = 2, Built Area = 3): 2
Pin Code of the City: 1100
Furnishing Status of the House (Unfurnished = 0, Semi-Furnished = 1, Furnished = 2): 1
Tenant Type (Bachelors = 1, Bachelors/Family = 2, Only Family = 3): 3
Number of bathrooms: 2
Predicted House Price =  [[34922.3]]
```

الملخص

هذه هي كيفية استخدام التعلم الآلي للتنبؤ بإيجار عقار سكني. مع البيانات المناسبة وتقنيات التعلم الآلي، تجد العديد من المنصات العقارية خيارات الإسكان وفقًا لميزانية العميل. آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بإيجار المنزل باستخدام التعلم الآلي باستخدام بايثون.

4) مدقق قوة كلمة المرور مع التعلم الآلي Password Strength Checker with Machine Learning

مدقق قوة كلمة المرور (Password Strength Checker) هو تطبيق يتحقق من مدى قوة كلمة المرور. تستخدم بعض مقاييس قوة كلمة المرور الشائعة خوارزميات التعلم الآلي للتنبؤ بقوة كلمة المرور الخاصة بك. لذلك، إذا كنت تريد معرفة كيفية استخدام التعلم الآلي للتحقق من قوة كلمة المرور، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على كيفية إنشاء مدقق قوة كلمة المرور باستخدام التعلم الآلي باستخدام بايثون.

كيفية إنشاء مدقق قوة كلمة المرور؟

يعمل مدقق قوة كلمة المرور من خلال فهم مجموعة الأرقام والأحرف والرموز الخاصة التي تستخدمهافي كلمة مرورك. يتم إنشاؤه من خلال تدريب نموذج التعلم الآلي على مجموعة بيانات مصنفة من مجموعات مختلفة من الأحرف والرموز الخاصة التي يستخدمها الأشخاصفي كلمات المرور. يتعلم النموذج من البيانات حول مجموعات الأحرف والرموز التي يمكن تصنيفها على أنها كلمة مرور صلبة أو ضعيفة.

لذلك لإنشاء تطبيق للتحقق من قوة كلمات المرور، نحتاج إلى مجموعة بيانات معنونة حول مجموعات مختلفة من الأحرف والرموز. لقد عثرت على مجموعة بيانات على Kaggle لتدريب نموذج التعلم الآلي للتنبؤ بقوة كلمة المرور. يمكننا استخدام تلك البيانات لهذه المهمة. يمكنك تنزيل مجموعة البيانات من هنا.

في القسم أدناه، سوف أطلعك على كيفية استخدام التعلم الآلي لإنشاء مدقق قوة كلمة المرور باستخدام بايثون.

مدقق قوة كلمة المرور باستخدام بايثون

لنبدأ باستيراد مكتبات بايثون ومجموعة البيانات اللازمة لإنشاء مدقق قوة كلمة المرور:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

data = pd.read_csv("data.csv", error_bad_lines=False)
print(data.head())
```

```
      password  strength
0     kzde5577         1
1     kino3434         1
2    visi7k1yr         1
3     megzy123         1
4  lamborghin1         1
```

تتكون مجموعة البيانات من عمودين؛ كلمة المرور (password) والقوة (strength).في عمود القوة:

0 يعني: قوة كلمة المرور ضعيفة (weak)؛

1 تعني: قوة كلمة المرور متوسطة (medium)؛

2 تعني: قوة كلمة المرور قوية (strong)؛

قبل المضي قدمًا، سأحول القيم 0 و 1 و 2في عمود القوة إلى قيم ضعيفة ومتوسطة وقوية:

```
data = data.dropna()
data["strength"] = data["strength"].map({0: "Weak ,"
" :1                                        Medium,"
" :2                                        Strong"})
print(data.sample(5))
```

```
           password strength
476676       xupet0n     Weak
112569   cdm06690669   Medium
267402  bluerose1291   Medium
237407    2298409uur   Medium
336018       jejien8     Weak
```

نموذج التنبؤ بقوة كلمة المرور

دعنا الآن ننتقل إلى تدريب نموذج التعلم الآلي للتنبؤ بقوة كلمة المرور. قبل أن نبدأفي إعداد النموذج، نحتاج إلى ترميز (tokenize) كلمات المرور لأننا نحتاج إلى النموذج للتعلم من مجموعات الأرقام والحروف والرموز للتنبؤ بقوة كلمة المرور. إذن، إليك كيف يمكننا ترميز البيانات وتقسيمها إلى مجموعات تدريب واختبار:

```
def word(password):
  character=[]
  for i in password:
    character.append(i)
  return character

x = np.array(data["password"])
y = np.array(data["strength"])

tdif = TfidfVectorizer(tokenizer=word)
x = tdif.fit_transform(x)
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
```

```
                    test_size=0.05,
                    random_state=42)
```

إليك الآن كيفية تدريب نموذج التصنيف للتنبؤ بقوة كلمة المرور:

```
model = RandomForestClassifier()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

```
0.956991816498417
```

الآن إليك كيف يمكننا التحقق من قوة كلمة المرور باستخدام النموذج المدرب:

```
import getpass
user = getpass.getpass("Enter Password: ")
data = tdif.transform([user]).toarray()
output = model.predict(data)
print(output)
```

```
Enter Password: ··········
['Strong']
```

الملخص

هذه هي الطريقة التي يمكنك بها استخدام التعلم الآلي لإنشاء مدقق قوة كلمة المرور باستخدام لغة برمجة بايثون. يعمل مدقق قوة كلمة المرور من خلال فهم مجموعة الأرقام والأحرف والرموز الخاصة التي تستخدمهافي كلمة مرورك. آمل أن تكون قد أحببت هذه المقالة حول إنشاء مدقق قوة كلمة المرور باستخدام التعلم الآلي باستخدام بايثون.

5) اكتشاف التعليقات غير المرغوب فيها باستخدام التعلم الآلي Spam Comments Detection with Machine Learning

يعني اكتشاف التعليقات غير المرغوب (Spam comments detection) فيها تصنيف التعليقات على أنها تعليقات غير مرغوب بها (spam) أو ليست تعليقات غير مرغوب بها (not spam). يُعد YouTube أحد الأنظمة الأساسية التي تستخدم التعلم الآلي لتصفية التعليقات غير المرغوب فيها تلقائيًا لحفظ منشئيها من التعليقات غير المرغوب فيها. إذا كنت تريد معرفة كيفية اكتشاف التعليقات غير المرغوب فيها باستخدام "التعلم الآلي"، فهذه المقالة مناسبة لك. في هذه المقالة، سوف آخذك خلال مهمة الكشف عن التعليقات غير المرغوب فيها باستخدام التعلم الآلي باستخدام بايثون.

اكتشاف التعليقات غير المرغوب فيها

يعد اكتشاف التعليقات غير المرغوب فيها مهمة تصنيف النص (text classification)في التعلم الآلي. التعليقات غير المرغوب فيها على منصات التواصل الاجتماعي هي نوع التعليقات المنشورة لإعادة توجيه المستخدم إلى حساب وسائط اجتماعية آخر أو موقع ويب أو أي جزء من المحتوى.

لاكتشاف التعليقات غير المرغوب فيها باستخدام التعلم الآلي، نحتاج إلى بيانات مصنفة لتعليقات غير المرغوب بها. لحسن الحظ، وجدت مجموعة بيانات على Kaggle حول تعليقات غير المرغوب بها على YouTube والتي ستكون مفيدة لمهمة اكتشاف التعليقات غير المرغوب فيها. يمكنك تنزيل مجموعة البيانات من هنا.

في القسم أدناه، ستتعلم كيفية اكتشاف التعليقات غير المرغوب فيها باستخدام التعلم الآلي باستخدام لغة برمجة بايثون.

اكتشاف التعليقات غير المرغوب فيها باستخدام بايثون

لنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import BernoulliNB

data = pd.read_csv("Youtube01-Psy.csv")
print(data.sample(5))
```

```
                            COMMENT_ID          AUTHOR  \
287    z13vhnh5ewvdyzh3o23bjz55lxbwjznor04    diego acosta
43     z12jvnua2tifirkvk23cfjtpxwmgxfch004   Didier Drogba
265    z13ucxdzemugi1v5n04ccjloko25drfb4js  Haley Harmicar
322    z13uffbajziyw5cfp23bwbw5auytzdl5b04   Juris Dumagan
89   z12pzpvbfl2igbwhe04cihtpuwymvr5gvsg0k     NstyIC Gold

                   DATE                                          CONTENT  \
287  2014-11-08T10:05:27  If I get 100 subscribers, I will summon Freddy...
43   2014-01-20T06:57:25  http://www.twitch.tv/jaroadc come follow and w...
265  2014-11-08T05:35:42  9 year olds be like, 'How does this have 2 bil...
322  2014-11-12T11:03:25            I think he was drunk during this :) x)
89   2014-11-03T20:41:23  Ching Ching ling long ding ring yaaaaaa Ganga ...

     CLASS
287      1
43       1
265      0
322      0
89       0
```

نحتاج فقط إلى عمود المحتوى (content column) وعمود الفئة (class column) من مجموعة البيانات لبقية المهمة. لذلك دعونا نحدد كلا العمودين ونتحرك أبعد من ذلك:

```
data = data[["CONTENT", "CLASS"]]
print(data.sample(5))
```

```
                                       CONTENT  CLASS
160  CHECK MY CHANNEL FOR MY NEW SONG 'STATIC'!! YO...      1
157              Follow me on Twitter @mscalifornia95       1
336  To everyone joking about how he hacked to get ...      0
329  FOLLOW MY COMPANY ON TWITTER  thanks.  https:/...      1
79   Hi there~I'm group leader of Angel, a rookie K...      1
```

يحتوي عمود الفئة على القيمتين 0 و 1. يشير 0 إلى أنه ليس تعليق غير مرغوب به، ويشير الرقم 1 إلى تعليق غير مرغوب به. ولجعل الأمر يبدو أفضل، سأستخدم تسميات "spam" و "not spam" بدلاً من 1 و 0:

```
data["CLASS"] = data["CLASS"].map({0: "Not Spam,"
                                   " :1  Spam Comment")}
print(data.sample(5))
```

```
                                       CONTENT         CLASS
161          Incmedia.org where the truth meets you.  Spam Comment
335  Hey guys can you check my YouTube channel I kn...  Spam Comment
134                              ❤ ❤ ❤ ❤ ❤❤❤❤       Not Spam
209  How can this music video get 2 billion views w...      Not Spam
45   ....subscribe.....  .....to my.........  .......  Spam Comment
```

تدريب نموذج التصنيف

دعنا الآن ننتقل إلى أبعد من ذلك من خلال تدريب نموذج التعلم الآلي لتصنيف التعليقات الى "spam" و "not spam". نظرًا لأن هذه المشكلة هي مشكلة تصنيف ثنائي، سأستخدم خوارزمية Bernoulli Naive Bayes لتدريب النموذج:

```
x = np.array(data["CONTENT"])
y = np.array(data["CLASS"])

cv = CountVectorizer()
x = cv.fit_transform(x)
xtrain, xtest, ytrain, ytest = train_test_split(x, y ,

test_size=0.2 ,

random_state=42(

model = BernoulliNB()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

```
0.9857142857142858
```

دعنا الآن نختبر النموذج من خلال تقديم تعليقات "spam" و"not spam"كإدخال:

```
sample = "Check this out: https://thecleverprogrammer.com "/
data = cv.transform([sample]).toarray()
print(model.predict(data))
```

```
['Spam Comment']
```

```
sample = "Lack of information "!
data = cv.transform([sample]).toarray()
print(model.predict(data))
```

```
['Not Spam']
```

هذه هي الطريقة التي يمكنك بها تدريب نموذج التعلم الآلي على مهمة اكتشاف البريد العشوائي باستخدام بايثون.

الملخص

يعني اكتشاف التعليقات غير المرغوب فيها تصنيف التعليقات على أنها تعليق غير مرغوب به أو ليست تعليق غير مرغوب به. التعليقات غير المرغوب فيها على منصات التواصل الاجتماعي هي نوع التعليقات المنشورة لإعادة توجيه المستخدم إلى حساب وسائط اجتماعية آخر أو موقع ويب أو أي جزء من المحتوى. آمل أن تكون قد أحببت هذه المقالة حول اكتشاف التعليقات غير المرغوب فيها باستخدام التعلم الآلي.

6) التنبؤ بطلب الطعام عبر الإنترنت باستخدام التعلم الآلي Online Food Order Prediction with Machine Learning

كان هناك طلب كبير على طلبات الطعام عبر الإنترنت (online food orders) بعد إدخال Swiggy و Zomatoفي السوق. تستخدم شركات توصيل الطعام عاداتك الشرائية لتسريع عملية التوصيل. يعد نظام التنبؤ بطلب الطعام أحد الأساليب المفيدة التي يمكن أن تستخدمها هذه الشركات لتسريع عملية التسليم بأكملها.في هذه المقالة، سأطلعك على مهمة التنبؤ بطلب الطعام عبر الإنترنت باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بطلب الطعام عبر الإنترنت: دراسة حالة

بعد تطبيق نظام توصيل الطعام الكامل عبر الإنترنت، ستحاول شركات مثل Swiggy و Zomato دائمًا تحسينها. الهدف الرئيسي لهذه الشركات تجاه عملائها هو توصيل الطعامفي الوقت المناسب. لتقديم الطعام بشكل أسرع، تحدد هذه الشركات المجالات التي يكون فيها الطلب على طلبات الطعام عبر الإنترنت مرتفعًا وتوظف المزيد من شركاء التوصيلفي تلك المواقع. يساعدفي توصيل الطعام بشكل أسرعفي المناطق التي يوجد بها المزيد من الطلبات.

تمتلك هذه الشركات الكثير من البيانات حول عملائها لدرجة أنهم يعرفون الآن عادات طلب الطعام لجميع العملاء. باستخدام هذه البيانات، يمكنهم أيضًا التنبؤ بما إذا كان العميل سيطلب مرة أخرى من تطبيقه أم لا. إنها تقنية جيدة لتحديد المناطق والعائلات وأنواع العملاء مع المزيد من فرص العمل.

آمل أن تكون قد فهمت الآن حالة استخدام أنظمة التنبؤ بطلب الطعام عبر الإنترنت.في القسم أدناه، سأطلعك على كيفية تدريب نموذج التعلم الآلي للتنبؤ بطلبات الطعام عبر الإنترنت من عميل معين.

التنبؤ بطلبات الطعام عبر الإنترنت باستخدام لغة بايثون

لنبدأ الآن بمهمة التنبؤ بطلب الطعام عبر الإنترنت باستخدام التعلم الآلي. يمكنك تنزيل مجموعة البيانات التي أستخدمها لهذه المهمة من هنا. لنبدأ باستيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import numpy as np
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
```

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style="whitegrid")

data = pd.read_csv("onlinefoods.csv")
print(data.head())
```

```
   Age  Gender Marital Status Occupation  Monthly Income  \
0   20  Female         Single    Student       No Income
1   24  Female         Single    Student  Below Rs.10000
2   22    Male         Single    Student  Below Rs.10000
3   22  Female         Single    Student       No Income
4   22    Male         Single    Student  Below Rs.10000

  Educational Qualifications  Family size  latitude  longitude  Pin code  \
0             Post Graduate            4   12.9766    77.5993    560001
1                  Graduate            3   12.9770    77.5773    560009
2             Post Graduate            3   12.9551    77.6593    560017
3                  Graduate            6   12.9473    77.5616    560019
4             Post Graduate            4   12.9850    77.5533    560010

  Output   Feedback Unnamed: 12
0    Yes   Positive         Yes
1    Yes   Positive         Yes
2    Yes  Negative          Yes
3    Yes   Positive         Yes
4    Yes   Positive         Yes
```

لذلك تحتوي مجموعة البيانات على معلومات مثل:

1. عمر العميل (Age).
2. جنس العميل (Gender).
3. الحالة الاجتماعية للعميل (Marital Status).
4. مهنة العميل (Occupation).
5. الدخل الشهري للعميل (Monthly Income).
6. المؤهل التعليمي للعميل (Educational Qualifications).
7. حجم الأسرة للعميل (Family size).
8. خطوط الطول والعرض لموقع العميل (longitude,latitude).
9. الرقم السري الخاص بمحل إقامة العميل (Pin code).

10. هل طلب العميل مرة أخرى (Output).

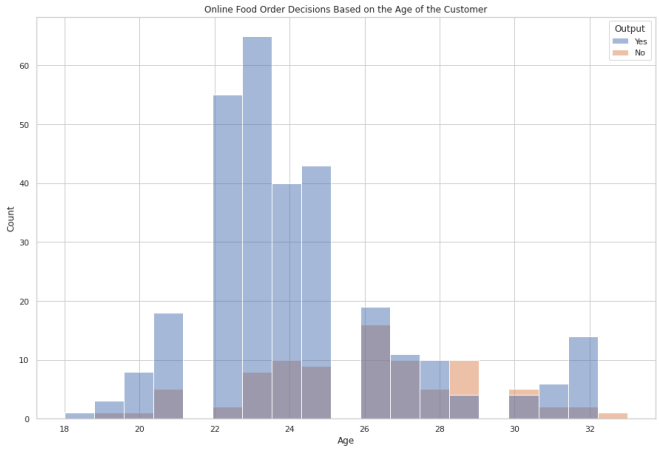
11. تعليقات (Feedback) على الطلب الأخير (إيجابي أو سلبي).

دعنا نلقي نظرة على المعلومات المتعلقة بجميع الأعمدةفي مجموعة البيانات:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 388 entries, 0 to 387
Data columns (total 13 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Age                        388 non-null    int64
 1   Gender                     388 non-null    object
 2   Marital Status             388 non-null    object
 3   Occupation                 388 non-null    object
 4   Monthly Income             388 non-null    object
 5   Educational Qualifications 388 non-null    object
 6   Family size                388 non-null    int64
 7   latitude                   388 non-null    float64
 8   longitude                  388 non-null    float64
 9   Pin code                   388 non-null    int64
 10  Output                     388 non-null    object
 11  Feedback                   388 non-null    object
 12  Unnamed: 12                388 non-null    object
dtypes: float64(2), int64(3), object(8)
memory usage: 39.5+ KB
None
```

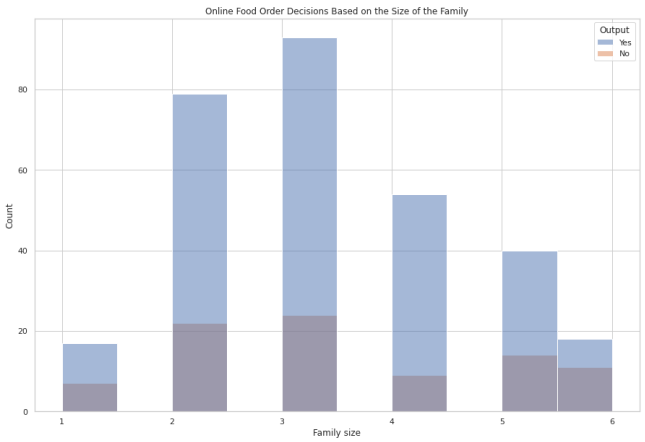
دعنا الآن ننتقل إلى تحليل هذه البيانات. سأبدأ بإلقاء نظرة على قرارات طلب الطعام عبر الإنترنت بناءً على عمر العميل:

```
plt.figure(figsize=(15, 10))
plt.title("Online Food Order Decisions Based on the Age of the Customer")
sns.histplot(x="Age", hue="Output", data=data)
plt.show()
```



يمكننا أن نرى أن الفئة العمرية من 22ـ25 طلبت الطعام مرارًا وتكرارًا. وهذا يعني أيضًا أن هذه الفئة العمرية هي هدف شركات توصيل الطعام عبر الإنترنت. دعنا الآن نلقي نظرة على قرارات طلب الطعام عبر الإنترنت بناءً على حجم عائلة العميل:

```
plt.figure(figsize=(15, 10))
plt.title("Online Food Order Decisions Based on the Size of
the Family")
sns.histplot(x="Family size", hue="Output", data=data)
plt.show()
```



تطلب العائلات المكونة من 2 و 3 أفراد الطعام كثيرًا. يمكن أن يكون هؤلاء رفقاءفي السكن أو أزواج أو أسرة مكونة من ثلاثة أفراد.

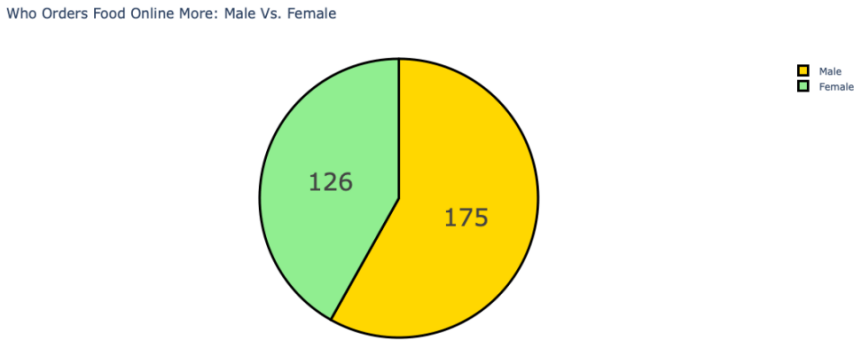
لننشئ مجموعة بيانات لجميع العملاء الذين طلبوا الطعام مرة أخرى:

```
buying_again_data = data.query("Output == 'Yes'")
print(buying_again_data.head())
```

```
   Age  Gender Marital Status Occupation  Monthly Income  \
0   20  Female         Single    Student       No Income
1   24  Female         Single    Student  Below Rs.10000
2   22    Male         Single    Student  Below Rs.10000
3   22  Female         Single    Student       No Income
4   22    Male         Single    Student  Below Rs.10000

  Educational Qualifications  Family size  latitude  longitude  Pin code  \
0             Post Graduate            4   12.9766    77.5993    560001
1                 Graduate             3   12.9770    77.5773    560009
2             Post Graduate            3   12.9551    77.6593    560017
3                 Graduate             6   12.9473    77.5616    560019
4             Post Graduate            4   12.9850    77.5533    560010

  Output   Feedback Unnamed: 12
0    Yes   Positive         Yes
1    Yes   Positive         Yes
2    Yes  Negative          Yes
3    Yes   Positive         Yes
4    Yes   Positive         Yes
```

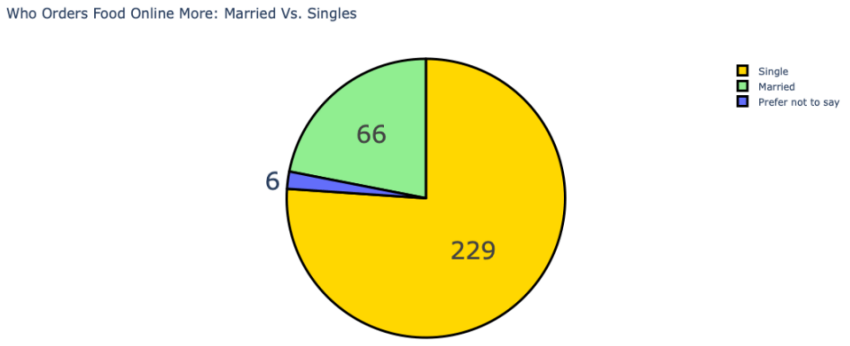
الآن دعونا نلقي نظرة على عمود الجنس. لنتعرف على من يطلب الطعام عبر الإنترنت:

```
gender = buying_again_data["Gender"].value_counts()
label = gender.index
counts = gender.values
colors = ['gold','lightgreen']

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Who Orders Food Online More:
Male Vs. Female')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3))(

fig.show()
```



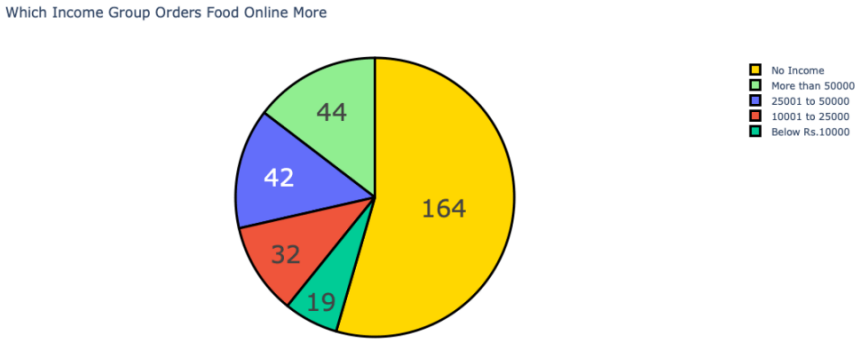
الآن دعونا نلقي نظرة على عمود الجنس. لنتعرف على من يطلب الطعام عبر الإنترنت:

```
marital = buying_again_data["Marital Status"].value_counts()
label = marital.index
counts = marital.values
colors = ['gold','lightgreen']

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Who Orders Food Online More:
Married Vs. Singles')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3)))
fig.show()
```



وفقًا للرقم أعلاه، 76.1٪ من العملاء الدائمين هم من العزاب. دعنا الآن نلقي نظرة على مجموعة دخل العملاء الذين طلبوا الطعام مرة أخرى:

```
income = buying_again_data["Monthly Income"].value_counts()
label = income.index
counts = income.values
colors = ['gold','lightgreen']

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Which Income Group Orders Food
Online More')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3)))
fig.show()
```



وفقًا للرقم أعلاه، لا يندرج 54٪ من العملاء تحت أي فئة دخل. يمكن أن يكونوا ربات بيوت أو طلاب.

دعنا الآن نجهز البيانات لمهمة تدريب نموذج التعلم الآلي. سأقوم هنا بتحويل جميع الميزات الفئوية (categorical features) إلى قيم عددية (numerical values):

```
data["Gender"] = data["Gender"].map({"Male": 1, "Female": 0})
data["Marital Status"] = data["Marital
Status"].map({"Married": 2,
"                      Single": 1,
"                      Prefer not to say": 0})
data["Occupation"] = data["Occupation"].map({"Student": 1,
"                    Employee": 2,
"                    Self Employeed": 3,
"                    House wife": 4})
data["Educational Qualifications"] = data["Educational
Qualifications"].map({"Graduate": 1,
"                                 Post Graduate": 2,
"                                 Ph.D": 3, "School": 4,
"                                 Uneducated": 5})
data["Monthly Income"] = data["Monthly Income"].map({"No
Income": 0,
 25001"                        to 50000": 5000,
"                      More than 50000": 7000,
 10001"                        to 25000": 25000,
"                      Below Rs.10000": 10000})
data["Feedback"] = data["Feedback"].map({"Positive": 1,
"Negative ": 0})
print(data.head())
```

	Age	Gender	Marital Status	Occupation	Monthly Income	\
0	20	0	1	1	0	
1	24	0	1	1	10000	
2	22	1	1	1	10000	
3	22	0	1	1	0	
4	22	1	1	1	10000	

	Educational Qualifications	Family size	latitude	longitude	Pin code	\
0	2	4	12.9766	77.5993	560001	
1	1	3	12.9770	77.5773	560009	
2	2	3	12.9551	77.6593	560017	
3	1	6	12.9473	77.5616	560019	
4	2	4	12.9850	77.5533	560010	

	Output	Feedback	Unnamed: 12
0	Yes	1	Yes
1	Yes	1	Yes
2	Yes	0	Yes
3	Yes	1	Yes
4	Yes	1	Yes

نموذج التنبؤ بطلب الطعام عبر الإنترنت

دعنا الآن ندرب نموذج التعلم الآلي للتنبؤ بما إذا كان العميل سيطلب مرة أخرى أم لا. سأبدأ بتقسيم البيانات إلى مجموعات تدريب واختبار:

```
#splitting data
from sklearn.model_selection import train_test_split
x = np.array(data[["Age", "Gender", "Marital Status",
"Occupation","
"        Monthly Income", "Educational Qualifications","
"        Family size", "Pin code", "Feedback([["
y = np.array(data[["Output"]])
```

دعنا الآن ندرب نموذج التعلم الآلي:

```
#training a machine learning model
from sklearn.ensemble import RandomForestClassifier
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                    test_size=0.10,
                    random_state=42(
model = RandomForestClassifier()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

```
0.9487179487179487
```

دعنا الآن نجهز نموذجًا لإدخال بيانات العميل ونتوقع ما إذا كان العميل سيطلب الطعام مرة أخرى أم لا:

```
print("Enter Customer Details to Predict If the Customer Will
Order Again")
a = int(input("Enter the Age of the Customer: "))
b = int(input("Enter the Gender of the Customer (1 = Male, 0 =
Female): "))
c = int(input("Marital Status of the Customer (1 = Single, 2 =
Married, 3 = Not Revealed): "))
d = int(input("Occupation of the Customer (Student = 1,
Employee = 2, Self Employeed = 3, House wife = 4): "))
e = int(input("Monthly Income: "))
f = int(input("Educational Qualification (Graduate = 1, Post
Graduate = 2, Ph.D = 3, School = 4, Uneducated = 5): "))
g = int(input("Family Size: "))
h = int(input("Pin Code: "))
i = int(input("Review of the Last Order (1 = Positive, 0 =
Negative): "))
features = np.array([[a, b, c, d, e, f, g, h, i]])
print("Finding if the customer will order again: ",
model.predict(features))
```

```
Enter Customer Details to Predict If the Customer Will Order Again
Enter the Age of the Customer: 22
Enter the Gender of the Customer (1 = Male, 0 = Female): 1
Marital Status of the Customer (1 = Single, 2 = Married, 3 = Not Revealed): 1
Occupation of the Customer (Student = 1, Employee = 2, Self Employeed = 3, House wife = 4): 1
Monthly Income: 0
Educational Qualification (Graduate = 1, Post Graduate = 2, Ph.D = 3, School = 4, Uneducated = 5): 1
Family Size: 3
Pin Code: 560010
Review of the Last Order (1 = Positive, 0 = Negative): 1

Finding if the customer will order again:  ['Yes']
```

الملخص

هذه هي الطريقة التي يمكنك بها توقع ما إذا كان العميل سيطلب الطعام عبر الإنترنت مرة أخرى أم لا. يعد نظام التنبؤ بطلب الطعام أحد الأساليب المفيدة التي يمكن لشركات توصيل الطعام استخدامها لتسريع عملية التسليم بأكملها. أتمنى أن تكون قد أحببت هذه المقالة حول توقع توصيل الطعام عبر الإنترنت باستخدام التعلم الآلي.

7) نظام توصية Instagram مع تعلم الآلة Instagram Recommendation System with Machine Learning

Instagram هو أحد تطبيقات الوسائط الاجتماعية الشائعة اليوم. يستخدم الناس Instagram لمشاركة الصور ومقاطع الفيديو. أحد أسباب شعبية Instagram هو نظام التوصية (recommendation system) الخاص به. يساعدفي جذب انتباه المستخدم لفترة أطول. لذلك، إذا كنت تريد معرفة كيفية استخدام التعلم الآلي للتوصية بمنشورات Instagram، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على كيفية إنشاء نظام توصية Instagram مع التعلم الآلي باستخدام بايثون.

كيف يعمل نظام توصيات Instagram؟

المنشور الذي تراه كمنشور مقترح على Instagram عندما تتصفح خلاصتك هو المكان الذي يستخدم فيه Instagram نظام توصية للتوصية بالمنشورات التي قد تهمك.

يستخدم Instagram أيضًا نظام التوصية لاقتراح المزيد من الحسابات لاتباعها، ولكن هنا سأطلعك على كيفية توصية نظام التوصيةفي Instagram بمزيد من المنشورات.

يوصى بالمنشورات المقترحة التي تراها على Instagram بناءً على أنشطتك على Instagram، مثل:

1. ما نوع الحسابات التي تتابعها، وما نوع المنشورات (posts) التي تتفاعل معها.
2. تلعب التسمية التوضيحية (caption) للمشاركات التي تتفاعل معها دورًافي اقتراح المزيد من المنشورات المشابهة.
3. كيف يتفاعل المستخدمون الآخرون الذين لديهم اهتمامات مماثلة لاهتماماتكفي المنشورات.

لذلك كانت هذه هي العوامل الأكثر شيوعًا لاقتراح المزيد من المنشورات على Instagram.في القسم أدناه، سأطلعك على كيفية إنشاء نظام توصية Instagram باستخدام التسميات التوضيحية لمنشورات Instagram.

نظام توصية Instagram باستخدام بايثون

يتم جمع مجموعة البيانات التي أستخدمها لإنشاء نظام توصية Instagram من حساب Instagram الخاص بي. يمكنك تنزيل مجموعة البيانات من هنا. دعنا الآن نستورد مكتبات بايثون ومجموعة البيانات اللازمة للبدء بهذه المهمة:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction import text
from sklearn.metrics.pairwise import cosine_similarity

data = pd.read_csv("Instagram data.csv")
print(data.head())
```

```
         Date  Impressions  From Home  From Hashtags  From Explore  \
0  2021-12-10         3920       2586           1028           619
1  2021-12-11         5394       2727           1838          1174
2  2021-12-12         4021       2085           1188             0
3  2021-12-13         4528       2700            621           932
4  2021-12-14         2518       1704            255           279

   From Other  Saves  Comments  Shares  Likes  Profile Visits  Follows  \
0          56     98         9       5    162              35        2
1          78    194         7      14    224              48       10
2         533     41        11       1    131              62       12
3          73    172        10       7    213              23        8
4          37     96         5       4    123               8        0

   Conversion Rate                                            Caption  \
0         5.714286  Here are some of the most important data visua...
1        20.833333  Here are some of the best data science project...
2        19.354839  Learn how to train a machine learning model an...
3        34.782609  Here's how you can write a Python program to d...
4         0.000000  Plotting annotations while visualizing your da...

                                        Hashtags
0  #finance #money #business #investing #investme...
1  #healthcare #health #covid #data #datascience ...
2  #data #datascience #dataanalysis #dataanalytic...
3  #python #pythonprogramming #pythonprojects #py...
4  #datavisualization #datascience #data #dataana...
```

سأختار فقط عمود التسمية التوضيحية (caption) وعلامات الهاشتاك (hashtags)لبقية المهمة:

```
data = data[["Caption", "Hashtags"]]
print(data.head())
```

```
                                        Caption  \
0  Here are some of the most important data visua...
1  Here are some of the best data science project...
2  Learn how to train a machine learning model an...
3  Here's how you can write a Python program to d...
4  Plotting annotations while visualizing your da...

                                       Hashtags
0  #finance #money #business #investing #investme...
1  #healthcare #health #covid #data #datascience ...
2  #data #datascience #dataanalysis #dataanalytic...
3  #python #pythonprogramming #pythonprojects #py...
4  #datavisualization #datascience #data #dataana...
```

سأستخدم هنا تشابه جيب التمام (cosine similarity)في التعلم الآلي للعثور على أوجه التشابه بين التسميات التوضيحية. بعد العثور على أوجه التشابه بين المنشورات، يمكننا التوصية بها للمستخدم بناءً على أوجه التشابه مع المنشور الذي تفاعل معه المستخدم للتو. فيما يلي كيف يمكننا التوصية بمنشورات Instagram:

```
captions = data["Caption"].tolist()
uni_tfidf = text.TfidfVectorizer(input=captions,
stop_words="english")
uni_matrix = uni_tfidf.fit_transform(captions)
uni_sim = cosine_similarity(uni_matrix)

def recommend_post(x):
  return ", ".join(data["Caption"].loc[x.argsort()[-5:-1]])

data["Recommended Post"] = [recommend_post(x) for x in
uni_sim]
print(data.head())
```

```
                                        Caption  \
0  Here are some of the most important data visua...
1  Here are some of the best data science project...
2  Learn how to train a machine learning model an...
3  Here's how you can write a Python program to d...
4  Plotting annotations while visualizing your da...

                                       Hashtags  \
0  #finance #money #business #investing #investme...
1  #healthcare #health #covid #data #datascience ...
2  #data #datascience #dataanalysis #dataanalytic...
3  #python #pythonprogramming #pythonprojects #py...
4  #datavisualization #datascience #data #dataana...

                                    Recommended Post
0  Here are some of the most important tools that...
1  Here are some of the best data science project...
2  Data Science Use Cases: Here's how Zomato is u...
3  Here's how to write a Python function to rever...
4  Practice these 90+ Data Science Projects For B...
```

دعنا الآن نلقي نظرة على توصيات منشور Instagram من مجموعة البيانات:

```
print(data["Recommended Post"][3])
```

```
Here’s how to write a Python function to reverse a string.,
To calculate the execution time of the program, we need to calculate the time taken by the program
from its initiation to the final result. Here’s how to calculate the execution time of a Python
program.,
Here’s how to calculate execution time of a Python program.,
Grouping anagrams is one of the popular questions in coding interviews. Here you will be given a list
of words, and you have to write an algorithm to group all the words which are anagrams of each other.
Here's how to group anagrams using Python.
[ ]
```

هذه هي الطريقة التي يمكنك بها التوصية بمنشورات Instagram بناءً على تسميات المنشورات.

الملخص

المنشور الذي تراه كمنشور مقترح على Instagram عند التمرير عبر خلاصتك هو المكان الذي يستخدم فيه Instagram نظام التوصية للتوصية بالمنشورات التي قد تهمك. يساعدفي جذب انتباه المستخدم لفترة أطول. أتمنى أن تكون قد أحببت هذه المقالة حول إنشاء نظام توصية Instagram مع التعلم الآلي باستخدام بايثون.

8) التنبؤ بدرجات الطالب مع التعلم الآلي Student Marks Prediction with Machine Learning

التنبؤ بدرجات الطالب (Student marks prediction) هي دراسة حالة علمية شائعة تستند إلى مشكلة الانحدار (regression). إنها مشكلة انحدار جيدة للمبتدئينفي علم البيانات (data science) حيث يسهل حلها وفهمها. لذلك إذا كنت تريد معرفة كيفية التنبؤ بعلامات الطالب باستخدام التعلم الآلي (machine learning)، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بدرجات الطلاب باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بدرجات الطالب (دراسة حالة)

يتم إعطاؤك بعض المعلومات حول الطلاب مثل:

1. عدد الكورسات التي اختاروها.
2. متوسط الوقت الذي يدرسه الطلاب يوميًا.
3. الدرجات التي حصل عليها الطلاب.

باستخدام هذه المعلومات، تحتاج إلى التنبؤ بدرجات الطلاب الآخرين. يمكنك تنزيل مجموعة البيانات من هنا.

التنبؤ بدرجات الطالب باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة توقع علامات الطلاب من Kaggle. لنبدأ الآن بهذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import numpy as np
import pandas as pd
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

data = pd.read_csv("Student_Marks.csv")
print(data.head(10))
```

```
   number_courses  time_study   Marks
0               3       4.508  19.202
1               4       0.096   7.734
2               4       3.133  13.811
3               6       7.909  53.018
4               8       7.811  55.299
5               6       3.211  17.822
6               3       6.063  29.889
7               5       3.413  17.264
8               4       4.410  20.348
9               3       6.173  30.862
```

لذلك لا يوجد سوى ثلاثة أعمدةفي مجموعة البيانات. عمود الدرجات (marks column) هو العمود المستهدف حيث يتعين علينا التنبؤ بدرجات الطالب.

الآن قبل المضي قدمًا، دعنا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

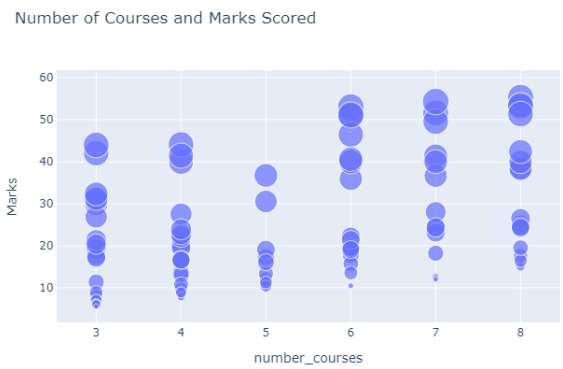
مجموعة البيانات جاهزة للاستخدام نظرًا لعدم وجود قيم خاليةفي البيانات. يوجد عمودفي البيانات يحتوي على معلومات حول عدد الكورسات التي اختارها الطلاب. لنلقِ نظرة على عدد قيم جميع قيم هذا العمود:

```
data["number_courses"].value_counts()
```

```
3    22
4    21
6    16
8    16
7    15
5    10
Name: number_courses, dtype: int64
```

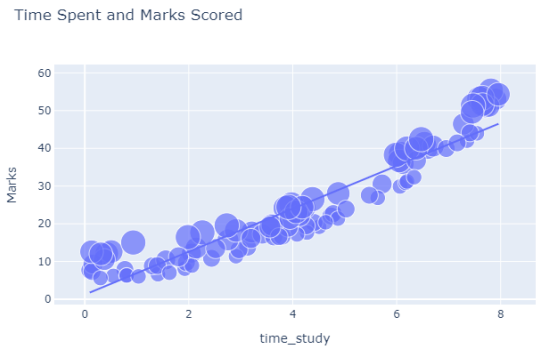
لذلك هناك ما لا يقل عن ثلاث كورسات كحد أدنى وثماني كورسات كحد أقصى اختارها الطلاب. دعنا نلقي نظرة على مخطط مبعثر (scatter plot) لمعرفة ما إذا كان عدد الكورسات التدريبية يؤثر على درجات الطالب:

```
figure = px.scatter(data_frame=data, x = "number_courses”,
                    y = "Marks", size = "time_study”,
                    title="Number of Courses and Marks
Scored”)
figure.show ()
```



وفقًا لتصوير البيانات أعلاه، يمكننا القول إن عدد الكورسات قد لا يؤثر على درجات الطالب إذا كان الطالب يدرس لمزيد من الوقت يوميًا. فلنلقِ نظرة على العلاقة بين الوقت الذي يدرس فيه يوميًا والدرجات التي سجلها الطالب:

```
figure = px.scatter(data_frame=data, x = "time_study"’’
        y = "Marks", size = "number_courses”,
        title="Time Spent and Marks Scored", trendline="ols”)
figure.show()
```



يمكنك أن ترى أن هناك علاقة خطية بين الوقت المدروس (time studied) والدرجات التي تم الحصول عليها. هذا يعني أنه كلما زاد الوقت الذي يقضيه الطلاب في الدراسة، كان بإمكانهم تحقيق نتائج أفضل.

دعنا الآن نلقي نظرة على العلاقة بين الدرجات التي سجلها الطلاب والعمودان الآخران في البيانات:

```
correlation = data.corr()
print(correlation["Marks"].sort_values(ascending=False))
```

```
Marks            1.000000
time_study       0.942254
number_courses   0.417335
Name: Marks, dtype: float64
```

لذا فإن العمود time_studied يكون أكثر ارتباطًا بعمود الدرجات أكثر من العمود الآخر.

نموذج التنبؤ بعلامات الطالب

دعنا الآن ننتقل إلى مهمة تدريب نموذج التعلم الآلي للتنبؤ بدرجات الطالب. هنا، سأبدأ أولاً بتقسيم البيانات إلى مجموعات تدريب واختبار:

```
x = np.array(data[["time_study", "number_courses"]])
y = np.array(data["Marks"])
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
```

```
                    test_size=0.2,
                    random_state=42)
```

الآن سأقوم بتدريب نموذج التعلم الآلي باستخدام خوارزمية الانحدار الخطي (linear regression):

```
model = LinearRegression()
model.fit(xtrain, ytrain)
model.score(xtest, ytest)
```

```
0.9459936100591212
```

دعنا الآن نختبر أداء نموذج التعلم الآلي هذا من خلال تقديم المدخلات بناءً على الميزات التي استخدمناها لتدريب النموذج والتنبؤ بدرجات الطالب:

```
#Features = [["time_study", "number_courses"]]
features = np.array([[3,4.508]])
model.predict(features)
```

```
array([22.30738483])
```

هذه هي الطريقة التي يمكنك بها التنبؤ بدرجات الطالب الذي لديه التعلم الآلي باستخدام بايثون.

الملخص

هذه هي الطريقة التي يمكنك بها حل مشكلة التنبؤ بدرجات الطلاب باستخدام التعلم الآلي. إنها مشكلة انحدار جيدة للمبتدئينفي علم البيانات حيث يسهل حلها وفهمها. آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بدرجات الطلاب باستخدام التعلم الآلي باستخدام بايثون.

9) كشف الاحتيال في المدفوعات عبر الإنترنت باستخدام التعلم الآلي Online Payments Fraud Detection with Machine Learning

ساعد إدخال أنظمة الدفع عبر الإنترنت كثيرًافي سهولة الدفع. لكنفي الوقت نفسه، زادت عمليات الاحتيالفي الدفع (payment frauds). يمكن أن تحدث عمليات الاحتيالفي الدفع عبر الإنترنت مع أي شخص يستخدم أي نظام دفع، خاصة أثناء إجراء المدفوعات باستخدام بطاقة الائتمان. هذا هو السببفي أن اكتشاف الاحتيالفي الدفع عبر الإنترنت مهم جدًا لشركات بطاقات الائتمان لضمان عدم تحصيل رسوم من العملاء مقابل المنتجات والخدمات التي لم يدفعوها مطلقًا. إذا كنت تريد معرفة كيفية اكتشاف عمليات الاحتيالفي الدفع عبر الإنترنت، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة الكشف عن الاحتيالفي المدفوعات عبر الإنترنت باستخدام التعلم الآلي باستخدام بايثون.

كشف الاحتيال في المدفوعات عبر الإنترنت باستخدام التعلم الآلي

لتحديد الاحتيالفي الدفع عبر الإنترنت باستخدام التعلم الآلي، نحتاج إلى تدريب نموذج التعلم الآلي لتصنيف المدفوعات الاحتيالية وغير الاحتيالية. لهذا، نحتاج إلى مجموعة بيانات تحتوي على معلومات حول الاحتيالفي الدفع عبر الإنترنت، حتى نتمكن من فهم نوع المعاملات التي تؤدي إلى الاحتيال. لهذه المهمة، جمعت مجموعة بيانات من Kaggle، والتي تحتوي على معلومات تاريخية حول المعاملات الاحتيالية التي يمكن استخدامها للكشف عن الاحتيالفي المدفوعات عبر الإنترنت. فيما يلي جميع الأعمدة من مجموعة البيانات التي أستخدمها هنا:

1. step: تمثل وحدة زمنية حيث تساوي الخطوة الواحدة ساعة واحدة.
1. type: نوع المعاملة (transaction) عبر الإنترنت.
2. amount: مبلغ المعاملة.
3. nameOrig: العميل الذي يبدأ المعاملة.
4. oldbalanceOrg: الرصيد قبل المعاملة.
5. newbalanceOrig: الرصيد بعد المعاملة.
6. nameDest: مستلم المعاملة.
7. oldbalanceDest: الرصيد المبدئي للمستلم قبل المعاملة.

8. newbalanceDest: الرصيد الجديد للمستلم بعد المعاملة.

9. isFraud: معاملة احتيال.

أتمنى أن تعرف الآن البيانات التي أستخدمهافي مهمة الكشف عن الاحتيالفي الدفع عبر الإنترنت. الآنفي القسم أدناه، سأشرح كيف يمكننا استخدام التعلم الآلي للكشف عن الاحتيالفي الدفع عبر الإنترنت باستخدام بايثون.

كشف الاحتيال للمدفوعات عبر الإنترنت باستخدام بايثون

سأبدأ هذه المهمة عن طريق استيراد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لهذه المهمة:

```
import pandas as pd
import numpy as np
data = pd.read_csv("credit card.csv")
print(data.head())
```

```
   step      type    amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
0     1   PAYMENT   9839.64  C1231006815       170136.0       160296.36
1     1   PAYMENT   1864.28  C1666544295        21249.0        19384.72
2     1  TRANSFER    181.00  C1305486145          181.0            0.00
3     1  CASH_OUT    181.00   C840083671          181.0            0.00
4     1   PAYMENT  11668.14  C2048537720        41554.0        29885.86

      nameDest  oldbalanceDest  newbalanceDest  isFraud  isFlaggedFraud
0  M1979787155             0.0             0.0        0               0
1  M2044282225             0.0             0.0        0               0
2   C553264065             0.0             0.0        1               0
3    C38997010         21182.0             0.0        1               0
4  M1230701703             0.0             0.0        0               0
```

الآن، دعنا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

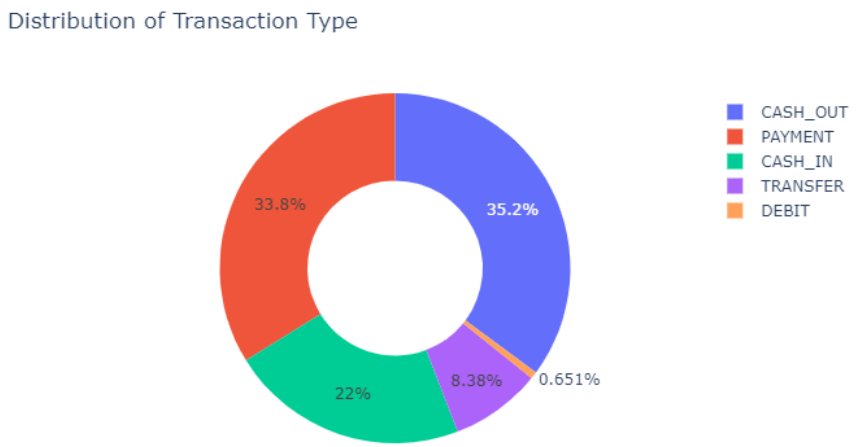
```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

لذلك لا تحتوي مجموعة البيانات هذه على أي قيم فارغة. قبل المضي قدمًا، دعنا الآن نلقي نظرة على نوع المعاملة المذكورةفي مجموعة البيانات:

```
 #Exploring transaction type
print(data.type.value_counts())
```

```
CASH_OUT    2237500
PAYMENT     2151495
CASH_IN     1399284
TRANSFER     532909
DEBIT         41432
Name: type, dtype: int64
```

```
type = data["type"].value_counts()
transactions = type.index
quantity = type.values

import plotly.express as px
figure = px.pie(data ,
                values=quantity ,
                names=transactions,hole = 0.5 ,

                title="Distribution of Transaction Type")
figure.show()
```



دعنا الآن نلقي نظرة على العلاقة بين ميزات البيانات مع عمود isFraud:

```
 #Checking correlation
correlation = data.corr()
```

```
print(correlation["isFraud"].sort_values(ascending=False))
```

```
isFraud           1.000000
amount            0.076688
isFlaggedFraud    0.044109
step              0.031578
oldbalanceOrg     0.010154
newbalanceDest    0.000535
oldbalanceDest   -0.005885
newbalanceOrig   -0.008148
Name: isFraud, dtype: float64
```

دعنا الآن نحول السمات الفئوية (categorical) إلى عددية (numerical). سأقوم هنا أيضًا بتحويل قيم عمود isFraud إلى تسميات No Fraud و Fraud للحصول على فهم أفضل للمخرجات:

```
data["type"] = data["type"].map({"CASH_OUT": 1, "PAYMENT": 2,
"               CASH_IN": 3, "TRANSFER": 4,
"               DEBIT": 5})
data["isFraud"] = data["isFraud"].map({0: "No Fraud", 1:
"Fraud"})
print(data.head())
```

```
   step  type    amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
0     1     2   9839.64  C1231006815       170136.0       160296.36
1     1     2   1864.28  C1666544295        21249.0        19384.72
2     1     4    181.00  C1305486145          181.0            0.00
3     1     1    181.00   C840083671          181.0            0.00
4     1     2  11668.14  C2048537720        41554.0        29885.86

      nameDest  oldbalanceDest  newbalanceDest   isFraud  isFlaggedFraud
0  M1979787155             0.0             0.0  No Fraud               0
1  M2044282225             0.0             0.0  No Fraud               0
2   C553264065             0.0             0.0     Fraud               0
3    C38997010         21182.0             0.0     Fraud               0
4  M1230701703             0.0             0.0  No Fraud               0
```

نموذج كشف الاحتيال للمدفوعات عبر الإنترنت

دعنا الآن ندرب نموذج تصنيف لتصنيف المعاملات الاحتيالية وغير الاحتيالية. قبل تدريب النموذج، سأقسم البيانات إلى مجموعات تدريب واختبار:

```
#splitting the data
from sklearn.model_selection import train_test_split
x = np.array(data[["type", "amount", "oldbalanceOrg",
"newbalanceOrig"]])
y = np.array(data[["isFraud"]])
```

الآن دعنا ندرب على نموذج الكشف عن الاحتيالفي المدفوعات عبر الإنترنت:

```
#training a machine learning model
from sklearn.tree import DecisionTreeClassifier
```

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.10, random_state=42)
model = DecisionTreeClassifier()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

```
0.9997391011878755
```

دعنا الآن نصنف ما إذا كانت المعاملة عبارة عن عملية احتيال أم لا من خلال تقديم معلومات حول معاملةفي النموذج:

```
#prediction
#features = [type, amount, oldbalanceOrg, newbalanceOrig]
features = np.array([[0.0,9000.60,9000.60,4]])
print(model.predict(features))
```

```
['Fraud']
```

الملخص

هذه هي الطريقة التي يمكننا بها اكتشاف الاحتيالفي المدفوعات عبر الإنترنت باستخدام التعلم الآلي باستخدام بايثون .يعد الكشف عن عمليات الاحتيالفي الدفع عبر الإنترنت أحد تطبيقات علم البياناتفي مجال التمويل. آمل أن تكون قد أحببت هذه المقالة حول اكتشاف الاحتيالفي المدفوعات عبر الإنترنت باستخدام التعلم الآلي باستخدام بايثون.

10) التنبؤ بإكرامية النادل مع التعلم الآلي Waiter Tips Prediction with Machine Learning

يعتمد تقديم اكرامية للنادل (Tipping waiter) لتقديم الطعام على العديد من العوامل مثل نوع المطعم وعدد الأشخاص الذين تتعامل معهم والمبلغ الذي تدفعه كفاتورتك وما إلى ذلك. يُعد تحليل اكرامية النادل أحد دراسات الحالة الشائعةفي علم البيانات حيث نحتاج إلى التنبؤ بالاكرامية التي تُعطى للنادل لتقديم الطعامفي المطعم. لذلك إذاكنت تريد معرفة كيفية حل دراسة الحالة هذه، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ باكرامية النادل مع التعلم الآلي باستخدام بايثون.

اكرامية النادل (دراسة حالة)

سجل خادم الطعامفي مطعم بيانات حول الاكراميات المعطاة للنادل لتقديم الطعام. البيانات التي سجلها خادم الطعام هي كما يلي:

1. total_bill: إجمالي الفاتورة بالدولار بمافي ذلك الضرائب.
2. tip: اكرامية تُعطى للنادل بالدولار.
3. sex: جنس الشخص الذي يدفع الفاتورة.
4. smoker: سواء كان الشخص مدخنًا أم لا.
5. day: يوم الأسبوع.
6. time: غداء أو عشاء.
7. size: عدد الأشخاص على الطاولة.

إذن هذه هي البيانات التي سجلها المطعم. بناءً على هذه البيانات، تتمثل مهمتنافي العثور على العوامل التي تؤثر على اكرامية النادل وتدريب نموذج تعلم الآلة للتنبؤ باكرامية النادل.

التنبؤ بإكرامية النادل باستخدام بايثون

لنبدأ الآن مهمة تحليل والتنبؤ بإكرامية النادل عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
data = pd.read_csv("tips.csv")
```

```
print(data.head())
```

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

فيما يلي الوصف الكامل لمجموعة البيانات هذه:

1. total_bill: إجمالي الفاتورة بالدولار بمافي ذلك الضريبة.
2. tip: اكرامية تُعطى للنادل بالدولار.
3. sex: جنس الشخص الذي يدفع الفاتورة.
4. smoker: سواء كان الشخص مدخنًا أم لا.
5. day: يوم الأسبوع.
6. time: غداء أو عشاء.
7. size: عدد الأشخاص.

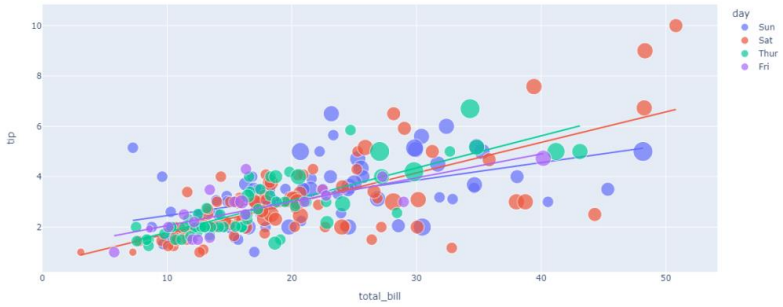
دعنا الآن ننتقل إلى الأمام من خلال تحليل جميع العوامل التي تؤثر على اكرامية النادل.

تحليل اكرامية النادل

دعونا نلقي نظرة على الاكرامية المقدمة للنادل وفقًا لما يلي:

1. إجمالي الفاتورة المدفوعة.
2. عدد الأشخاص على الطاولة.
3. ويوم الأسبوع.

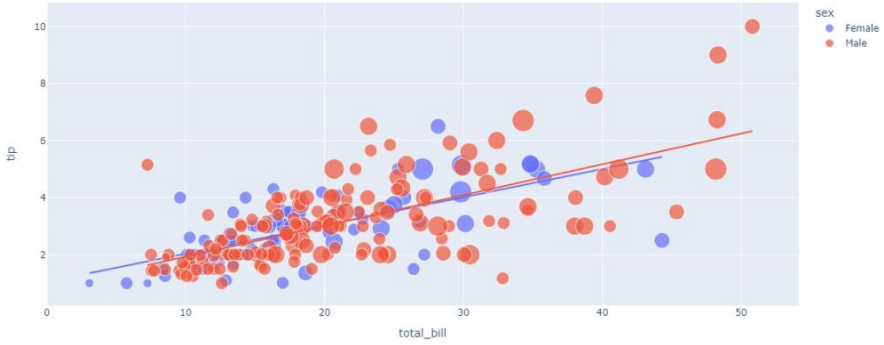
```
figure = px.scatter(data_frame = data, x="total_bill”,
        y="tip", size="size", color= "day", trendline="ols”)
figure.show()
```



الآن دعنا نلقي نظرة على الاكرامية المعطاة للنادل وفقًا لما يلي:

1. إجمالي الفاتورة المدفوعة.
2. عدد الأشخاص على الطاولة.
3. وجنس الشخص الذي يدفع الفاتورة.

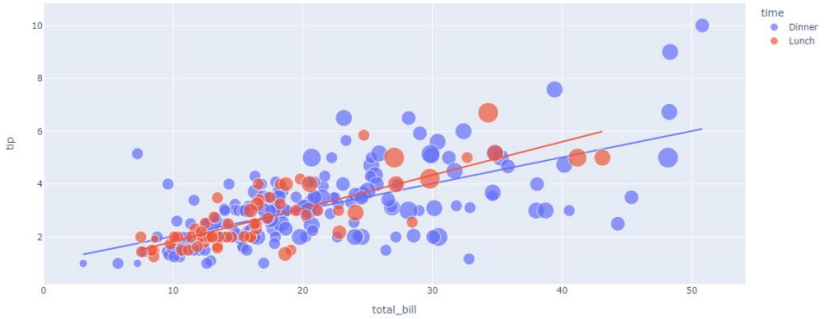
```
figure = px.scatter(data_frame = data, x="total_bill”,
        y="tip", size="size", color= "sex", trendline="ols”)
figure.show()
```



الآن دعنا نلقي نظرة على الاكرامية المعطاة للنادل وفقًا لما يلي:

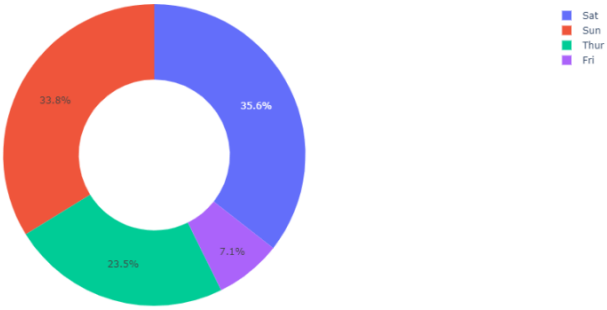
1. إجمالي الفاتورة المدفوعة.
2. عدد الأشخاص على الطاولة.
3. ووقت الوجبة.

```
figure = px.scatter(data_frame = data, x="total_bill”,
        y="tip", size="size", color= "time", trendline="ols”)
figure.show()
```



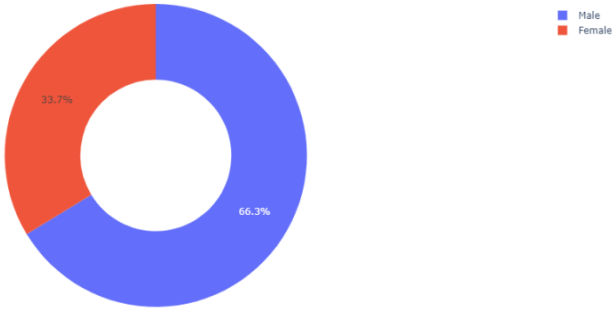
دعنا الآن نرى الاكرامية المقدمة للنادل وفقًا للأيام لمعرفة اليوم الذي يتم فيه تقديم معظم الاكراميات للنادل:

```
figure = px.pie(data,
     values='tip',
     names='day',hole = 0.5)
figure.show()
```



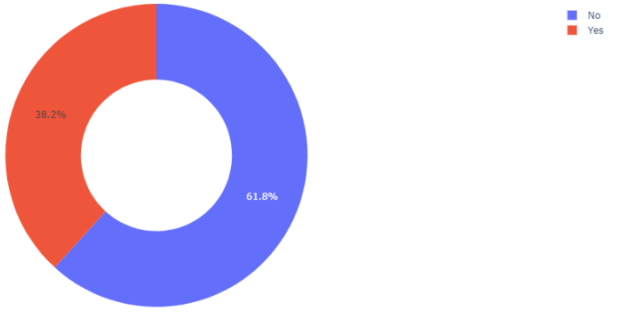
وفقًا للرسم أعلاه، يتم تقديم معظم الاكراميات في أيام السبت إلى النادلين. لنلقِ الآن نظرة على عدد الاكراميات التي تُعطى للنادل حسب جنس الشخص الذي يدفع الفاتورة لمعرفة من الذي يعطي النادلين أكثر من غيرهم:

```
figure = px.pie(data,
     values='tip',
     names='sex',hole = 0.5)
figure.show()
```



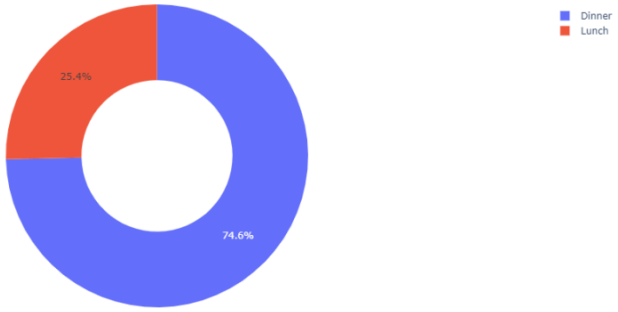
وفقًا للتصوير أعلاه، يتم تقديم معظم الاكراميات من قبل الرجال. دعنا الآن نرى ما إذا كان المدخن يقدم اكراميات أكثر أم أنه غير مدخن:

```
figure = px.pie(data,
     values='tip',
     names='smoker',hole = 0.5)
figure.show()
```



وفقًا للتصوير أعلاه، فإن اكراميات غير المدخنين هي أكثر من المدخنين. دعنا الآن نرى ما إذا كان يتم تقديم معظم الاكراميات أثناء الغداء أو العشاء:

```
figure = px.pie(data,
     values='tip',
     names='time',hole = 0.5)
figure.show()
```



وفقًا للتصوير أعلاه، يعطى اكرامية للنادل أكثر أثناء العشاء.

هذه هي الطريقة التي يمكننا بها تحليل جميع العوامل التي تؤثر على اكرامية النادل. الآنفي القسم أدناه، سوف آخذكفي جولة حول كيفية تدريب نموذج التعلم الآلي لمهمة التنبؤ باكرامية النادل.

نموذج التنبؤ بإكرامية النادل

قبل تدريب نموذج التنبؤ بإكرامية النادل، سأقوم ببعض تحويل البيانات عن طريق تحويل القيم الفئوية (categorical) إلى قيم عددية (numerical):

```
data["sex"] = data["sex"].map({"Female": 0, "Male": 1})
data["smoker"] = data["smoker"].map({"No": 0, "Yes": 1})
data["day"] = data["day"].map({"Thur": 0, "Fri": 1, "Sat": 2,
"Sun": 3})
data["time"] = data["time"].map({"Lunch": 0, "Dinner": 1})
data.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	0	0	3	1	2
1	10.34	1.66	1	0	3	1	3
2	21.01	3.50	1	0	3	1	3
3	23.68	3.31	1	0	3	1	2
4	24.59	3.61	0	0	3	1	4

الآن سأقسم البيانات إلى مجموعات تدريب واختبار:

```
x = np.array(data[["total_bill", "sex", "smoker", "day",
"time", "size"]])
y = np.array(data["tip"])

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y ,

test_size=0.2 ,random_state=42)
```

الآن فيما يلي كيفية تدريب نموذج التعلم الآلي لمهمة التنبؤ بإكرامية النادل باستخدام بايثون:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(xtrain, ytrain)
```

دعنا الآن نختبر أداء هذا النموذج من خلال إعطاء مدخلات لهذا النموذج وفقًا للميزات التي استخدمناها لتدريب هذا النموذج:

```
#features = [[total_bill, "sex", "smoker", "day", "time",
"size"]]
features = np.array([[4,1,0,0,1,24.50]])
model.predict(features)
```

```
array([3.73742609])
```

الملخص

هذه هي الطريقة التي يمكنك بها توقع اكرامية النادل باستخدام التعلم الآلي باستخدام بايثون . يعد تحليل اكرامية النادل أحد دراسات الحالة الشائعةفي مجال علوم البيانات حيث نحتاج إلى التنبؤ بالإكرامية التي يتم تقديمها للنادل لتقديم الطعامفي مطعم. آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بإكرامية النادل مع التعلم الآلي باستخدام بايثون.

11) تجميع أنواع الموسيقى باستخدام التعلم الآلي Clustering Music Genres with Machine Learning

التجميع (Clustering) هو أسلوب تعلم آلي لتجميع نقاط البيانات التي تتميز بميزات محددة. تجميع أنواع الموسيقى هو مهمة تجميع الموسيقى بناءً على أوجه التشابهفي خصائصها الصوتية. إذا كنت تريد معرفة كيفية إجراء تحليل المجموعات على أنواع الموسيقى، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة تجميع أنواع الموسيقى مع التعلم الآلي باستخدام بايثون.

تجميع أنواع الموسيقى (بيان المشكلة)

لكل شخص ذوق مختلففي الموسيقى. لا يمكننا تحديد نوع الموسيقى التي يحبها الشخص بمجرد معرفة أسلوب حياته أو هواياته أو مهنته. لذلك يصعب على تطبيقات دفق الموسيقى التوصية بالموسيقى لشخص ما. ولكن إذا عرفنا نوع الأغاني التي يستمع إليها الشخص يوميًا، فيمكننا إيجاد أوجه تشابهفي جميع ملفات الموسيقى والتوصية بموسيقى مماثلة للشخص.

هذا هو المكان الذي يأتي فيه التحليل العنقودي (cluster analysis) لأنواع الموسيقى. هنا تحصل على مجموعة بيانات من الأغاني الشعبية على Spotify، والتي تحتوي على أسماء الفنانين والموسيقى مع جميع الخصائص الصوتية لكل موسيقى. هدفك هو تجميع أنواع الموسيقى بناءً على أوجه التشابهفي خصائصها الصوتية.

يمكنك تنزيل مجموعة البيانات من هنا.

تجميع أنواع الموسيقى باستخدام لغة بايثون

آمل أن تكون قد فهمت بيان المشكلة المذكور أعلاه حول تجميع أنواع الموسيقى مع التعلم الآلي. لنبدأ الآن بهذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
from sklearn import cluster

data = pd.read_csv("Spotify-2000.csv")
print(data.head())
```

```
   Index                   Title             Artist            Top Genre  \
0      1                 Sunrise        Norah Jones      adult standards
1      2             Black Night        Deep Purple           album rock
2      3          Clint Eastwood           Gorillaz  alternative hip hop
3      4           The Pretender       Foo Fighters    alternative metal
4      5  Waitin' On A Sunny Day  Bruce Springsteen         classic rock

   Year  Beats Per Minute (BPM)  Energy  Danceability  Loudness (dB)  \
0  2004                     157      30            53            -14
1  2000                     135      79            50            -11
2  2001                     168      69            66             -9
3  2007                     173      96            43             -4
4  2002                     106      82            58             -5

   Liveness  Valence Length (Duration)  Acousticness  Speechiness  Popularity
0        11       68               201            94            3          71
1        17       81               207            17            7          39
2         7       52               341             2           17          69
3         3       37               269             0            4          76
4        10       87               256             1            3          59
```

يمكنك رؤية جميع أعمدة مجموعة البياناتفي الإخراج أعلاه. يحتوي على جميع الميزات الصوتية للموسيقى التي تكفي للعثور على أوجه التشابه. قبل المضي قدمًا، سأقوم بإسقاط عمود الفهرس (index column)، لأنه لا فائدة منه:

```
data = data.drop("Index", axis=1)
```

دعنا الآن نلقي نظرة على الارتباط (correlation) بين جميع ميزات الصوتفي مجموعة البيانات:

```
print(data.corr())
```

```
                        Year  Beats Per Minute (BPM)    Energy  \
Year                1.000000                0.012570  0.147235
Beats Per Minute (BPM)  0.012570                1.000000  0.156644
Energy                  0.147235                0.156644  1.000000
Danceability            0.077493               -0.140602  0.139616
Loudness (dB)           0.343764                0.092927  0.735711
Liveness                0.019017                0.016256  0.174118
Valence                -0.166163                0.059653  0.405175
Acousticness           -0.132946               -0.122472 -0.665156
Speechiness             0.054097                0.085598  0.205865
Popularity             -0.158962               -0.003181  0.103393

                       Danceability  Loudness (dB)  Liveness   Valence  \
Year                       0.077493       0.343764  0.019017 -0.166163
Beats Per Minute (BPM)    -0.140602       0.092927  0.016256  0.059653
Energy                     0.139616       0.735711  0.174118  0.405175
Danceability               1.000000       0.044235 -0.103063  0.514564
Loudness (dB)              0.044235       1.000000  0.098257  0.147041
Liveness                  -0.103063       0.098257  1.000000  0.050667
Valence                    0.514564       0.147041  0.050667  1.000000
Acousticness              -0.135769      -0.451635 -0.046206 -0.239729
Speechiness                0.125229       0.125090  0.092594  0.107102
Popularity                 0.144344       0.165527 -0.111978  0.095911
```

	Acousticness	Speechiness	Popularity
Year	-0.132946	0.054097	-0.158962
Beats Per Minute (BPM)	-0.122472	0.085598	-0.003181
Energy	-0.665156	0.205865	0.103393
Danceability	-0.135769	0.125229	0.144344
Loudness (dB)	-0.451635	0.125090	0.165527
Liveness	-0.046206	0.092594	-0.111978
Valence	-0.239729	0.107102	0.095911
Acousticness	1.000000	-0.098256	-0.087604
Speechiness	-0.098256	1.000000	0.111689
Popularity	-0.087604	0.111689	1.000000

التحليل العنقودي للميزات الصوتية

سأستخدم الآن خوارزمية التجميع K-mean للعثور على أوجه التشابه بين جميع ميزات الصوت. ثم سأضيف مجموعاتفي مجموعة البيانات بناءً على أوجه التشابه التي وجدناها. لذلك دعونا ننشئ مجموعة بيانات جديدة لجميع خصائص الصوت ونجري تحليل المجموعات باستخدام خوارزمية K-mean clustering:

```
data2 = data[["Beats Per Minute (BPM)", "Loudness (dB,"(
"      Liveness", "Valence", "Acousticness,"
"      Speechiness[["

from sklearn.preprocessing import MinMaxScaler
for i in data.columns:
  MinMaxScaler(i)

from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=10)
clusters = kmeans.fit_predict(data2)
```

سأضيف الآن المجموعات كما تنبأت خوارزمية التجميع K-mean إلى مجموعة البيانات الأصلية:

```
data["Music Segments"] = clusters
MinMaxScaler(data["Music Segments"])
data["Music Segments"] = data["Music Segments"].map({1:
"Cluster 1", 2:
"  Cluster 2", 3: "Cluster 3", 4: "Cluster 4", 5: "Cluster 5",
":6  Cluster 6", 7: "Cluster 7", 8: "Cluster 8",
":9  Cluster 9", 10: "Cluster 10"})
```

دعنا الآن نلقي نظرة على مجموعة البيانات ذات المجموعات(clusters):

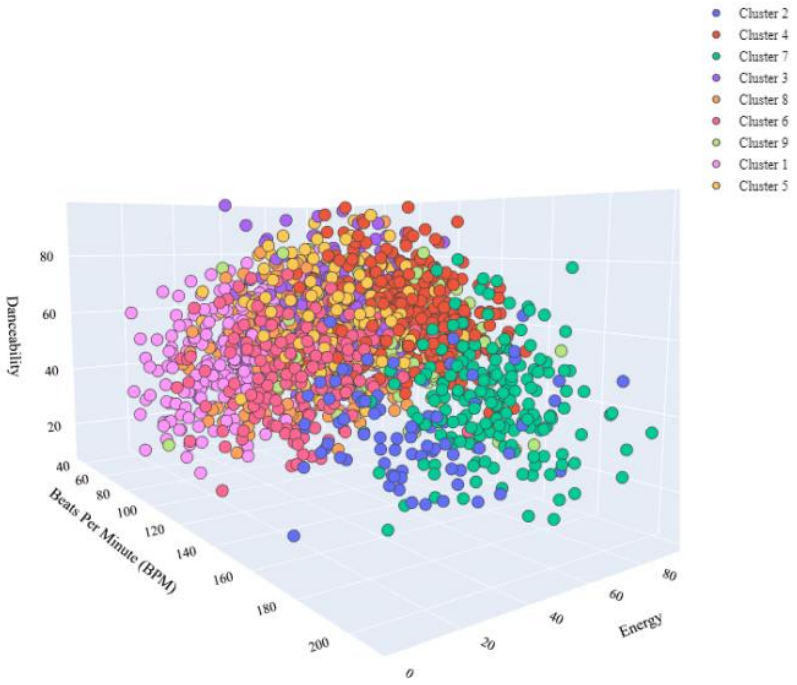
```
print(data.head())
```

```
                    Title             Artist           Top Genre  Year  \
0                Sunrise        Norah Jones      adult standards  2004
1             Black Night        Deep Purple          album rock  2000
2          Clint Eastwood           Gorillaz  alternative hip hop  2001
3           The Pretender       Foo Fighters   alternative metal  2007
4  Waitin' On A Sunny Day  Bruce Springsteen         classic rock  2002

   Beats Per Minute (BPM)  Energy  Danceability  Loudness (dB)  Liveness  \
0                     157      30            53            -14        11
1                     135      79            50            -11        17
2                     168      69            66             -9         7
3                     173      96            43             -4         3
4                     106      82            58             -5        10

   Valence  Length (Duration)  Acousticness  Speechiness  Popularity  \
0       68                201            94            3          71
1       81                207            17            7          39
2       52                341             2           17          69
3       37                269             0            4          76
4       87                256             1            3          59

  Music Segments
0      Cluster 1
1      Cluster 6
2      Cluster 2
3      Cluster 2
4      Cluster 3
```

الآن دعنا نتخيل المجموعات بناءً على بعض ميزات الصوت :

```
import plotly.graph_objects as go
PLOT = go.Figure()
for i in list(data["Music Segments"].unique()):


  PLOT.add_trace(go.Scatter3d(x = data[data["Music Segments"]==
i]['Beats Per Minute (BPM,['(
              y = data[data["Music Segments"] == i]['Energy'],
              z = data[data["Music Segments"] ==
i]['Danceability']          ,
              mode = 'markers',marker_size = 6,
marker_line_width = 1,
              name = str(i)))
PLOT.update_traces(hovertemplate='Beats Per Minute (BPM): %{x}
<br>Energy: %{y} <br>Danceability: %{z}')


PLOT.update_layout(width = 800, height = 800, autosize = True,
showlegend = True,
        scene = dict(xaxis=dict(title = 'Beats Per Minute
(BPM)', titlefont_color = 'black'),
              yaxis=dict(title = 'Energy', titlefont_color =
'black'),
```

```
            zaxis=dict(title = 'Danceability',
titlefont_color = 'black'),(
        font = dict(family = "Gilroy", color  = 'black', size =
12))
```



هذه هي الطريقة التي يمكننا بها إجراء التحليل العنقودي لأنواع الموسيقى باستخدام التعلم الآلي.

الملخص

هذه هي الطريقة التي يمكنك بها إجراء التحليل العنقودي لأنواع الموسيقى باستخدام التعلم الآلي باستخدام بايثون. تجميع أنواع الموسيقى هو مهمة تجميع الموسيقى بناءً على أوجه التشابهفي ميزات الصوت الخاصة بها. آمل أن تكون قد أحببت هذه المقالة حول تجميع أنواع الموسيقى مع التعلم الآلي.

12) التنبؤ بأسعار العملات المشفرة باستخدام التعلم الآلي Cryptocurrency Price Prediction with Machine Learning

يجب أن تكون قد سمعت أو استثمرتفي أي عملة مشفرة مرة (cryptocurrency) واحدةفي حياتك. إنها وسيلة تبادل رقمية مشفرة وغير مركزية. كثير من الناس يستخدمون العملات المشفرة كشكل من أشكال الاستثمار لأنها تعطي عوائد كبيرة حتىفي فترة قصيرة. تعد Bitcoin وEthereum و Binance Coin من بين العملات المشفرة الشائعة اليوم. إذا كنت تريد معرفة كيفية التنبؤ بالأسعار المستقبلية لأي عملة مشفرة باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة التنبؤ بأسعار العملات المشفرة (cryptocurrency price prediction) مع التعلم الآلي باستخدام بايثون.

التنبؤ بأسعار العملات المشفرة باستخدام التعلم الآلي

يعد التنبؤ بسعر العملات المشفرة أحد دراسات الحالة الشائعةفي مجتمع علوم البيانات. لا تعتمد أسعار الأسهم والعملات المشفرة فقط على عدد الأشخاص الذين يشترونها أو يبيعونها. اليوم، يعتمد التغييرفي أسعار هذه الاستثمارات أيضًا على التغييراتفي السياسات المالية للحكومة فيما يتعلق بأي عملة مشفرة. إن مشاعر الناس تجاه عملة مشفرة معينة أو شخصية معينة تصادق بشكل مباشر أو غير مباشر على عملة مشفرة تؤدي أيضًا إلى عمليات شراء وبيع ضخمة لعملة مشفرة معينة، مما يؤدي إلى حدوث تغييرفي الأسعار.

باختصار، يؤدي الشراء والبيع إلى تغيير سعر أي عملة مشفرة، لكن اتجاهات البيع والشراء تعتمد على العديد من العوامل. لا يمكن استخدام التعلم الآلي للتنبؤ بأسعار العملات المشفرة إلافي المواقف التي تتغير فيها الأسعار بسبب الأسعار التاريخية التي يراها الناس قبل شراء وبيع عملاتهم المشفرة. لذلك،في القسم أدناه، سأطلعك على كيفية توقع أسعار البيتكوين (وهي واحدة من أكثر العملات المشفرة شيوعًا) خلال الثلاثين يومًا القادمة.

التنبؤ بأسعار العملات المشفرة باستخدام بايثون

سأبدأ مهمة التنبؤ بأسعار العملات المشفرة عن طريق استيراد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها. لهذه المهمة، سأجمع أحدث بيانات أسعار البيتكوين من Yahoo Finance، باستخدام yfinance API. سيساعدك هذافي جمع أحدث البياناتفي كل مرة تقوم فيها بتشغيل هذا الكود:

```
import pandas as pd
import yfinance as yf
```

```
import datetime
from datetime import date, timedelta
today = date.today()

d1 = today.strftime("%Y-%m-%d")
end_date = d1
d2 = date.today() - timedelta(days=730)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2

data = yf.download('BTC-USD,'
         start=start_date,
         end=end_date,
         progress=False(
data["Date"] = data.index
data = data[["Date", "Open", "High", "Low", "Close", "Adj
Close", "Volume"]]
data.reset_index(drop=True, inplace=True)
print(data.head())
```

```
        Date         Open         High  ...        Close    Adj Close       Volume
0 2019-12-28  7289.031250  7399.041016  ...  7317.990234  7317.990234  21365673026
1 2019-12-29  7317.647461  7513.948242  ...  7422.652832  7422.652832  22445257702
2 2019-12-30  7420.272949  7454.824219  ...  7292.995117  7292.995117  22874131672
3 2019-12-31  7294.438965  7335.290039  ...  7193.599121  7193.599121  21167946112
4 2020-01-01  7194.892090  7254.330566  ...  7200.174316  7200.174316  18565664997

[5 rows x 7 columns]
```

في الكود أعلاه، قمت بجمع أحدث بيانات أسعار البيتكوين على مدار الـ 730 يومًا الماضية، ثم أعددتها لأي مهمة تتعلق بعلم البيانات. الآن، دعنا نلقي نظرة على شكل مجموعة البيانات هذه لمعرفة ما إذا كنا نعمل مع 730 صفًا أم لا:

```
data.shape
```

```
(731, 7)
```

لذلك تحتوي مجموعة البيانات على 731 صفًا، حيث يحتوي الصف الأول على أسماء كل عمود. الآن دعنا نتخيل التغييرفي أسعار البيتكوين حتى اليوم باستخدام مخطط الشموع (candlestick chart):

```
import plotly.graph_objects as go
figure = go.Figure(data=[go.Candlestick(x=data["Date,["
                                        open=data["Open"] ,
                                        high=data["High"],
                                        low=data["Low"] ,
                                        close=data["Close"]([(
figure.update_layout(title = "Bitcoin Price Analysis ,"
                     xaxis_rangeslider_visible=False(
figure.show()
```

يحتوي عمود الإغلاق (Close column)في مجموعة البيانات على القيم التي نحتاج إلى توقعها. لذلك، دعنا نلقي نظرة على ارتباط جميع الأعمدةفي البيانات المتعلقة بعمود الإغلاق:

```
correlation = data.corr()
print(correlation["Close"].sort_values(ascending=False))
```

```
Adj Close    1.000000
Close        1.000000
High         0.998933
Low          0.998740
Open         0.997557
Volume       0.334698
Name: Close, dtype: float64
```

نموذج التنبؤ بأسعار العملات المشفرة

يعتمد توقع الأسعار المستقبلية للعملات المشفرة على مشكلة تحليل السلاسل الزمنية (Time series analysis). تعد مكتبة AutoTS في بايثون واحدة من أفضل المكتبات لتحليل السلاسل الزمنية. لذلك سأستخدم هنا مكتبة AutoTS للتنبؤ بأسعار البيتكوين للأيام الثلاثين القادمة:

```
from autots import AutoTS
model = AutoTS(forecast_length=30, frequency='infer',
ensemble='simple')
model = model.fit(data, date_col='Date', value_col='Close',
id_col=None)
prediction = model.predict()
forecast = prediction.forecast
print(forecast)
```

```
                Close
2021-12-28  57865.012345
2021-12-29  54259.592685
2021-12-30  53794.634938
2021-12-31  54365.964301
2022-01-01  55371.531945
2022-01-02  57220.503886
2022-01-03  57132.487546
```

```
2022-01-05  58376.081818
2022-01-06  59931.323291
2022-01-07  60168.816716
2022-01-08  60617.974204
2022-01-09  58785.722512
2022-01-10  55180.302852
2022-01-11  54715.345105
2022-01-12  55286.674468
2022-01-13  56292.242112
2022-01-14  58141.214053
2022-01-15  58053.197713
2022-01-16  58942.437232
2022-01-17  59296.791985
2022-01-18  60852.033458
2022-01-19  61089.526883
2022-01-20  61538.684371
2022-01-21  59706.432679
2022-01-22  56101.013019
2022-01-23  55636.055272
2022-01-24  56207.384635
2022-01-25  57212.952279
2022-01-26  59061.924220
```

هذه هي الطريقة التي يمكنك بها استخدام التعلم الآلي للتنبؤ بسعر أي عملة مشفرة باستخدام لغة برمجة بايثون.

الملخص

ينتج عن الشراء والبيع تغييرفي سعر أي عملة مشفرة، لكن اتجاهات البيع والشراء تعتمد على العديد من العوامل. لا يمكن استخدام التعلم الآلي للتنبؤ بأسعار العملات المشفرة إلافي المواقف التي تتغير فيها الأسعار بسبب الأسعار التاريخية التي يراها الناس قبل شراء وبيع عملاتهم المشفرة. آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بأسعار العملات المشفرة باستخدام التعلم الآلي باستخدام بايثون.

13) التنبؤ بسعر السهم باستخدام التعلم الآلي Stock Price Prediction using Machine Learning

يعد التنبؤ بسوق الأوراق المالية (stock market prediction) أحد أهم تطبيقات التعلم الآلي في مجال التمويل.في هذه المقالة، سوف نخوضفي مشروع بسيط لعلم البيانات حول التنبؤ بأسعار الأسهم باستخدام لغة بايثون للتعلم الآلي.

في نهاية هذه المقالة، ستتعلم كيفية التنبؤ بأسعار الأسهم باستخدام نموذج الانحدار الخطي (Linear Regression model) من خلال تطبيق لغة برمجة بايثون.

التنبؤ بسعر السهم

لقد كان توقع سوق الأوراق المالية لعنة وهدف المستثمرين منذ نشأته. يتم تداول مليارات الدولارات يوميًافي البورصة، وخلف كل دولار يوجد مستثمر يأملفي تحقيق ربح بطريقة أو بأخرى.

الشركات بأكملها ترتفع وتنخفض يوميًا اعتمادًا على سلوك السوق. إذا كان المستثمر قادرًا على التنبؤ بدقة بحركات السوق، فإنه يقدم وعدًا محيرًا بالثروة والتأثير.

اليوم، يكسب الكثير من الناس المال من البقاءفي المنزلفي التداولفي سوق الأوراق المالية. إنها نقطة إضافية بالنسبة لك إذا كنت تستخدم خبرتكفي سوق الأوراق المالية ومهارات التعلم الآلي الخاصة بك لمهمة التنبؤ بسعر السهم.

دعنا نرى كيفية توقع أسعار الأسهم باستخدام التعلم الآلي ولغة برمجة بايثون. سأبدأ هذه المهمة عن طريق استيراد جميع مكتبات بايثون الضرورية التي نحتاجها لهذه المهمة:

```
import numpy as np
import pandas as pd
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

تحضير البيانات

في القسم أعلاه، بدأت مهمة التنبؤ بسعر السهم عن طريق استيراد مكتبات بايثون. سأكتب الآن دالة من شأنها إعداد مجموعة البيانات حتى نتمكن من ملاءمتها بسهولةفي نموذج الانحدار الخطي:

```
def prepare_data(df,forecast_col,forecast_out,test_size):
    label = df[forecast_col].shift(-forecast_out) #creating
new column called label with the last 5 rows are nan
    X = np.array(df[[forecast_col]]) #creating the feature
array
```

```
    X = preprocessing.scale(X) #processing the feature array
    X_lately = X[-forecast_out:] #creating the column i want
to use later in the predicting method
    X = X[:-forecast_out] # X that will contain the training
and testing
    label.dropna(inplace=True) #dropping na values
    y = np.array(label)  # assigning Y
    X_train, X_test, Y_train, Y_test = train_test_split(X, y,
test_size=test_size, random_state=0) #cross validation

    response = [X_train,X_test , Y_train, Y_test , X_lately]
    return response
```

يمكنك بسهولة فهم الدالة المذكورة أعلاه لأنني قد دونت طريقة عمل كل سطر خطوة بخطوة.

الآن الشيء التالي الذي يجب فعله هو قراءة البيانات:

```
df = pd.read_csv("prices.csv")
df = df[df.symbol == "GOOG"]
```

نحتاج الآن إلى إعداد ثلاث متغيرات إدخال كما تم إعدادها بالفعل في الدالة التي تم إنشاؤها في القسم أعلاه. نحتاج إلى إعلان متغير إدخال مع ذكر العمود الذي نريد التنبؤ به. المتغير التالي الذي نحتاج إلى الإعلان عنه هو إلى أي مدى نريد التنبؤ.

والمتغير الأخير الذي نحتاج إلى الإعلان عنه هو المقدار الذي يجب أن يكون حجم مجموعة الاختبار. الآن دعنا نعلن عن جميع المتغيرات:

```
forecast_col = 'close'
forecast_out = 5
test_size = 0.2
```

تطبيق تعلم الآلة للتنبؤ بأسعار الأسهم

الآن سأقسم البيانات وملائمة (fit) نموذج الانحدار الخطي:

```
X_train, X_test, Y_train, Y_test , X_lately
=prepare_data(df,forecast_col,forecast_out,test_size);
#calling the method were the cross validation and data
preperation is in
learner = LinearRegression() #initializing linear regression
model

learner.fit(X_train,Y_train) #training the linear regression
model
```

الآن دعونا نتنبأ بالمخرجات ونلقي نظرة على أسعار الأسهم:

```
score=learner.score(X_test,Y_test)#testing the linear
regression model
forecast= learner.predict(X_lately) #set that will contain the
forecasted data
response={}#creting json object
response['test_score']=score
response['forecast_set']=forecast
print(response)
```

{'test_score': 0.9481024935723803, 'forecast_set': array([786.54352516, 788.13020371, 781.84159626, 779.65508615, 769.04187979])}

الملخص

إذن هذه هي الطريقة التي يمكننا بها التنبؤ بأسعار الأسهم باستخدام التعلم الآلي. آمل أن تكون قد أحببت هذه المقالة حول تنبؤات أسعار الأسهم بايثون مع التعلم الآلي من خلال تطبيق نموذج الانحدار الخطي.

14) التنبؤ بالنجاة من سرطان الثدي مع التعلم الآلي Breast Cancer Survival Prediction with Machine Learning

سرطان الثدي (Breast cancer) هو أحد أنواع السرطانات التي تبدأفي الثدي. يحدث عند النساء، لكن يمكن أن يصاب الرجال أيضًا بسرطان الثدي. إنه السبب الرئيسي الثاني للوفاة عند النساء. نظرًا لأن استخدام البياناتفي الرعاية الصحية أمر شائع جدًا اليوم، يمكننا استخدام التعلم الآلي للتنبؤ بما إذا كان المريض سينجو من مرض مميت مثل سرطان الثدي أم لا. لذلك إذا كنت تريد معرفة كيفية التنبؤ بنجاة مريضة بسرطان الثدي، فهذه المقالة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بالنجاة من سرطان الثدي مع التعلم الآلي باستخدام بايثون.

التنبؤ بالنجاة من سرطان الثدي مع التعلم الآلي

لديك مجموعة بيانات تضم أكثر من 400 مريض بسرطان الثدي خضعن لعملية جراحية لعلاج سرطان الثدي. فيما يلي معلومات عن جميع الأعمدةفي مجموعة البيانات:

1. Patient_ID: معرف المريض.
2. Age: عمر المريض.
3. Gender: جنس المريض.
4. Protein1, Protein2, Protein3, Protein4: مستويات التعبير.
5. Tumor_Stage: مرحلة سرطان الثدي لدى المريضة.
6. Histology: تسلل سرطان الأقنية Infiltrating Ductal Carcinoma ، تسلل سرطاني فصيصي Infiltration Lobular Carcinoma ، سرطانة مخاطية Mucinous Carcinoma.
7. ER status: إيجابي / سلبي.
8. PR status: إيجابي / سلبي.
9. HER2 status: إيجابي / سلبي.

10. Surgery_type: استئصال الكتلة الورمية Lumpectomy ، استئصال الثدي البسيط Simple Mastectomy ، استئصال الثدي الجذري المعدل Modified Radical Mastectomy ، أخرى.

11. DateofSurgery: تاريخ الجراحة.

12. DateofLast_Visit: تاريخ آخر زيارة للمريض.

13. Patient_Status: حي / ميت.

لذلك، باستخدام مجموعة البيانات هذه، تتمثل مهمتنافي التنبؤ بما إذاكانت مريضة سرطان الثدي ستنجو أم لا بعد الجراحة.

آمل أن يكون لديك نظرة عامة على مجموعة البيانات التي نستخدمها لمهمة التنبؤ بالبقاء على قيد الحياة من سرطان الثدي. تم جمع مجموعة البيانات هذه من Kaggle. يمكنك تنزيل مجموعة البيانات هذه من هنا. الآن ،في القسم أدناه، سأوجهك خلال مهمة التنبؤ بالبقاء على قيد الحياة من سرطان الثدي باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بالنجاة من سرطان الثدي باستخدام لغة بايثون

سأبدأ مهمة التنبؤ بالبقاء على قيد الحياة من سرطان الثدي عن طريق استيراد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها:

```
import pandas as pd
import numpy as np
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

data = pd.read_csv("BRCA.csv")
print(data.head())
```

```
     Patient_ID   Age  Gender  Protein1  Protein2  Protein3  Protein4  \
0  TCGA-D8-A1XD  36.0  FEMALE  0.080353   0.42638   0.54715  0.273680
1  TCGA-EW-A1OX  43.0  FEMALE -0.420320   0.57807   0.61447 -0.031505
2  TCGA-A8-A079  69.0  FEMALE  0.213980   1.31140  -0.32747 -0.234260
3  TCGA-D8-A1XR  56.0  FEMALE  0.345090  -0.21147  -0.19304  0.124270
4  TCGA-BH-A0BF  56.0  FEMALE  0.221550   1.90680   0.52045 -0.311990

  Tumour_Stage                    Histology ER status PR status HER2 status  \
0          III  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
1           II            Mucinous Carcinoma  Positive  Positive    Negative
2          III  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
3           II  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
4           II  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
```

```
                Surgery_type Date_of_Surgery Date_of_Last_Visit  \
0  Modified Radical Mastectomy       15-Jan-17          19-Jun-17
1                  Lumpectomy       26-Apr-17          09-Nov-18
2                       Other       08-Sep-17          09-Jun-18
3  Modified Radical Mastectomy       25-Jan-17          12-Jul-17
4                       Other       06-May-17          27-Jun-19

  Patient_Status
0          Alive
1           Dead
2          Alive
3          Alive
4           Dead
```

دعونا نلقي نظرة على ما إذا كانت أعمدة مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا :

```
print(data.isnull().sum())
```

```
Patient_ID            7
Age                   7
Gender                7
Protein1              7
Protein2              7
Protein3              7
Protein4              7
Tumour_Stage          7
Histology             7
ER status             7
PR status             7
HER2 status           7
Surgery_type          7
Date_of_Surgery       7
Date_of_Last_Visit    24
Patient_Status        20
dtype: int64
```

إذن، تحتوي مجموعة البيانات هذه على بعض القيم الخاليةفي كل عمود، وسوف أسقط هذه القيم الفارغة:

```
data = data.dropna()
```

دعنا الآن نلقي نظرة على الإحصاءات حول أعمدة هذه البيانات:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 317 entries, 0 to 333
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Patient_ID          317 non-null    object
 1   Age                 317 non-null    float64
 2   Gender              317 non-null    object
 3   Protein1            317 non-null    float64
 4   Protein2            317 non-null    float64
 5   Protein3            317 non-null    float64
```

```
 6   Protein4           317 non-null    float64
 7   Tumour_Stage       317 non-null    object
 8   Histology          317 non-null    object
 9   ER status          317 non-null    object
 10  PR status          317 non-null    object
 11  HER2 status        317 non-null    object
 12  Surgery_type       317 non-null    object
 13  Date_of_Surgery    317 non-null    object
 14  Date_of_Last_Visit 317 non-null    object
 15  Patient_Status     317 non-null    object
dtypes: float64(5), object(11)
memory usage: 42.1+ KB
```

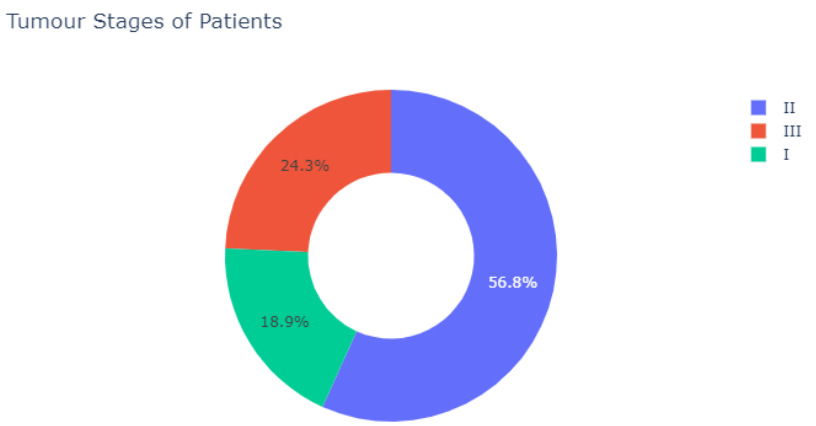
تم العثور على سرطان الثديفي الغالب عند الإناث، لذلك دعونا نلقي نظرة على عمود الجنس (Gender column) لمعرفة عدد الإناث والذكور هناك:

```
print(data.Gender.value_counts())
```

```
FEMALE    313
MALE        4
Name: Gender, dtype: int64
```

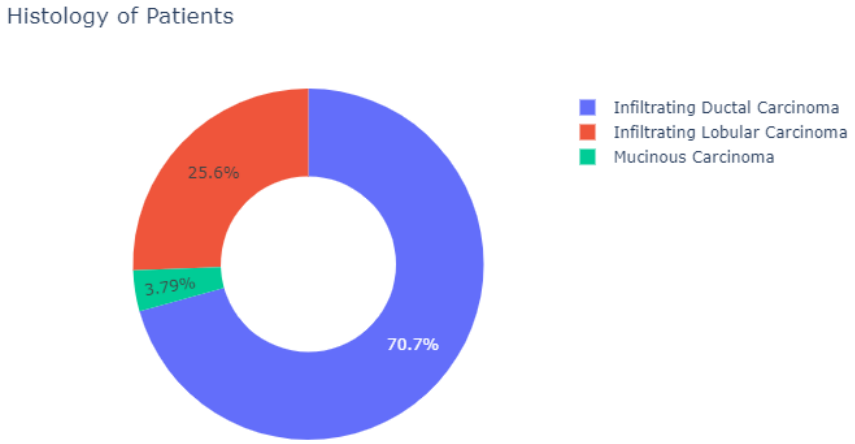
كما هو متوقع، فإن نسبة الإناث أكثر من الذكورفي عمود الجنس. الآن دعونا نلقي نظرة على مرحلة الورم (stage of tumor) لدى المرضى:

```
#Tumour Stage
stage = data["Tumour_Stage"].value_counts()
transactions = stage.index
quantity = stage.values

figure = px.pie(data,
     values=quantity,
     names=transactions,hole = 0.5,
     title="Tumour Stages of Patients("
figure.show()
```



لذلك معظم المرضىفي المرحلة الثانية. الآن دعونا نلقي نظرة على أنسجة (histology) مرضى سرطان الثدي. (علم الأنسجة هو وصف للورم بناءً على مدى تشوه الخلايا والأنسجة السرطانية تحت المجهر ومدى سرعة نمو وانتشار السرطان):

```
#Histology
histology = data["Histology"].value_counts()
transactions = histology.index
quantity = histology.values
figure = px.pie(data,
     values=quantity,
     names=transactions,hole = 0.5,
     title="Histology of Patients")
figure.show()
```



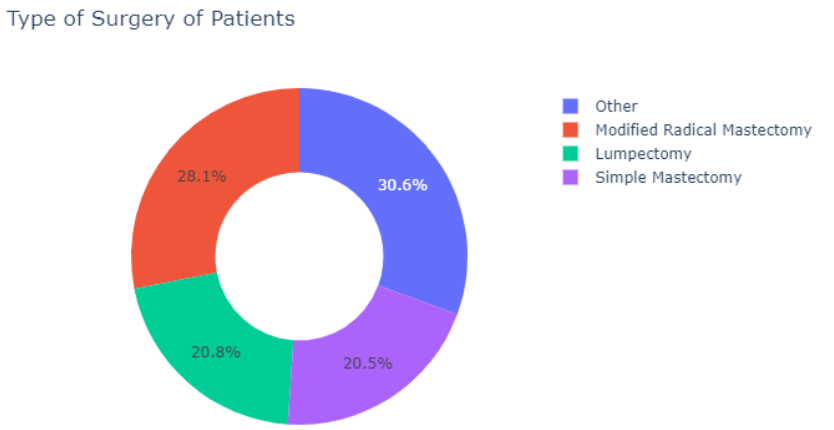
دعنا الآن نلقي نظرة على قيم ER status وPR status وHER2 status للمرضى:

```
#ER status
print(data["ER status"].value_counts())
#PR status
print(data["PR status"].value_counts())
#HER2 status
print(data["HER2 status"].value_counts())
```

```
Positive    317
Name: ER status, dtype: int64
Positive    317
Name: PR status, dtype: int64
Negative    288
Positive     29
Name: HER2 status, dtype: int64
```

الآن دعونا نلقي نظرة على نوع العمليات الجراحية (type of surgeries) التي أجريت للمرضى:

```
#Surgery_type
surgery = data["Surgery_type"].value_counts()
transactions = surgery.index
quantity = surgery.values
figure = px.pie(data,
     values=quantity,
     names=transactions,hole = 0.5,
     title="Type of Surgery of Patients")
figure.show()
```



لذلك اكتشفنا البيانات، تحتوي مجموعة البيانات على الكثير من الميزات الفئوية (categorical features). لاستخدام هذه البيانات لتدريب نموذج التعلم الآلي، نحتاج إلى تحويل قيم جميع الأعمدة الفئوية. إليك كيف يمكننا تحويل قيم الميزات الفئوية:

```
data["Tumour_Stage"] = data["Tumour_Stage"].map({"I": 1, "II":
2, "III": 3})
data["Histology"] = data["Histology"].map({"Infiltrating
Ductal Carcinoma": 1,
"                 Infiltrating Lobular Carcinoma": 2, "Mucinous
Carcinoma": 3({
data["ER status"] = data["ER status"].map({"Positive": 1})
data["PR status"] = data["PR status"].map({"Positive": 1})
data["HER2 status"] = data["HER2 status"].map({"Positive": 1,
"Negative": 2})
data["Gender"] = data["Gender"].map({"MALE": 0, "FEMALE": 1})
data["Surgery_type"] = data["Surgery_type"].map({"Other": 1,
"Modified Radical Mastectomy": 2,
"                   Lumpectomy": 3, "Simple Mastectomy": 4({
print(data.head())
```

```
     Patient_ID   Age  Gender  Protein1  Protein2  Protein3  Protein4  \
0  TCGA-D8-A1XD  36.0       1  0.080353   0.42638   0.54715  0.273680
1  TCGA-EW-A1OX  43.0       1 -0.420320   0.57807   0.61447 -0.031505
2  TCGA-A8-A079  69.0       1  0.213980   1.31140  -0.32747 -0.234260
3  TCGA-D8-A1XR  56.0       1  0.345090  -0.21147  -0.19304  0.124270
4  TCGA-BH-A0BF  56.0       1  0.221550   1.90680   0.52045 -0.311990

   Tumour_Stage  Histology  ER status  PR status  HER2 status  Surgery_type  \
0             3          1          1          1            2             2
1             2          3          1          1            2             3
2             3          1          1          1            2             1
3             2          1          1          1            2             2
4             2          1          1          1            2             1

  Date_of_Surgery Date_of_Last_Visit Patient_Status
0       15-Jan-17          19-Jun-17          Alive
1       26-Apr-17          09-Nov-18           Dead
2       08-Sep-17          09-Jun-18          Alive
3       25-Jan-17          12-Jul-17          Alive
4       06-May-17          27-Jun-19           Dead
```

نموذج التنبؤ بالنجاة من سرطان الثدي

يمكننا الآن الانتقال إلى تدريب نموذج التعلم الآلي للتنبؤ ببقاء مريض سرطان الثدي على قيد الحياة. قبل تدريب النموذج، نحتاج إلى تقسيم البيانات إلى مجموعة تدريب واختبار:

```
#Splitting data
x = np.array(data[['Age', 'Gender', 'Protein1', 'Protein2',
'Protein3','Protein4',
'        Tumour_Stage', 'Histology', 'ER status', 'PR status',
'        HER2 status', 'Surgery_type']])
y = np.array(data[['Patient_Status']])
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.10, random_state=42)
```

إليك الآن كيف يمكننا تدريب نموذج التعلم الآلي:

```
model = SVC()
model.fit(xtrain, ytrain)
```

دعنا الآن ندخل جميع الميزات التي استخدمناها لتدريب نموذج التعلم الآلي هذا والتنبؤ بما إذا كان المريض سينجو من سرطان الثدي أم لا:

```
#Prediction
#features = [['Age', 'Gender', 'Protein1', 'Protein2',
'Protein3','Protein4', 'Tumour_Stage', 'Histology', 'ER
status', 'PR status', 'HER2 status', 'Surgery_type']]
features = np.array ([[36.0, 1, 0.080353, 0.42638, 0.54715,
0.273680, 3, 1, 1, 1, 2, 2,]])
print(model.predict(features))
```

```
['Alive']
```

الملخص

هذه هي الطريقة التي يمكننا بها استخدام التعلم الآلي لمهمة التنبؤ بالبقاء على قيد الحياة من سرطان الثدي. نظرًا لأن استخدام البياناتفي الرعاية الصحية أمر شائع جدًا اليوم، يمكننا استخدام التعلم الآلي للتنبؤ بما إذا كان المريض سينجو من مرض مميت مثل سرطان الثدي أم لا. آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بالبقاء على قيد الحياة من سرطان الثدي مع التعلم الآلي باستخدام بايثون.

15) التنبؤ بوفيات Covid-19 باستخدام التعلم الآلي Covid-19 Deaths Prediction with Machine Learning

يعد Covid-19 أحد أكثر الفيروسات فتكًا التي سمعتها على الإطلاق. الطفراتفي Covid-19 تجعله إما أكثر فتكًا أو معديًا. لقد رأينا الكثير من الوفيات بسبب Covid-19 بينما كانت هناك موجة أعلى من الحالات. يمكننا استخدام البيانات التاريخية الخاصة بحالات Covid-19 والوفيات للتنبؤ بعدد الوفياتفي المستقبل. لذلك إذا كنت تريد معرفة كيفية التنبؤ بوفيات Covid-19 باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بوفيات Covid-19 مع التعلم الآلي باستخدام بايثون.

التنبؤ بوفيات كوفيد-19(دراسة حالة)

لقد حصلت على مجموعة بيانات لـ Covid-19في الهند من 30 يناير 2020 إلى 18 يناير 2022. تحتوي مجموعة البيانات على معلومات حول الحالات والوفيات المؤكدة يوميًا. فيما يلي جميع أعمدة مجموعة البيانات:

1. Date: يحتوي على تاريخ السجل.
2. Date_YMD: يحتوي على التاريخ بتنسيق Year-Month-Day.
3. Daily Confirmed: يحتوي على الحالات المؤكدة اليومية لـ Covid-19.
4. Daily Deceased: يحتوي على الوفيات اليومية بسبب Covid-19.

تحتاج إلى استخدام هذه البيانات التاريخية لحالات covid-19 والوفيات للتنبؤ بعدد الوفيات خلال الثلاثين يومًا القادمة. يمكنك تنزيل مجموعة البيانات هذه من هنا.

التنبؤ بوفيات Covid-19 باستخدام بايثون

آمل أن تكون قد فهمت الآن بيان المشكلة المذكور أعلاه. سأقوم الآن باستيراد جميع مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لمهمة التنبؤ بوفيات covid-19:

```
import pandas as pd
import numpy as np
data = pd.read_csv("COVID19 data for overall INDIA.csv")
print(data.head())
```

	Date	Date_YMD	Daily Confirmed	Daily Deceased
0	30 January 2020	2020-01-30	1	0
1	31 January 2020	2020-01-31	0	0
2	1 February 2020	2020-02-01	0	0
3	2 February 2020	2020-02-02	1	0
4	3 February 2020	2020-02-03	1	0

قبل المضي قدمًا، دعنا نلقي نظرة سريعة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

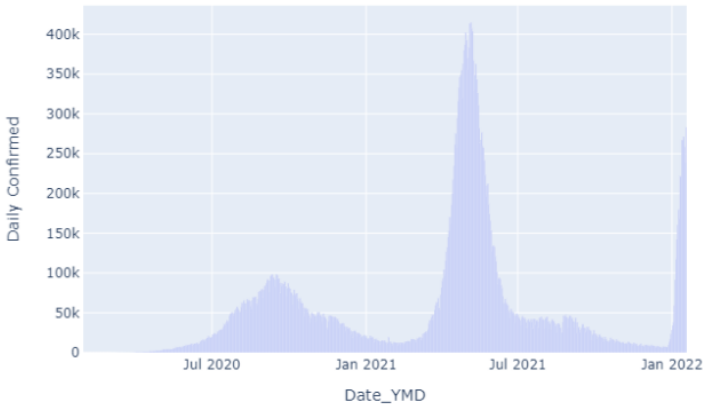
```
Date             0
Date_YMD         0
Daily Confirmed  0
Daily Deceased   0
dtype: int64
```

لسنا بحاجة إلى عمود التاريخ (date column)، لذا دعنا نسقط هذا العمود من مجموعة البيانات الخاصة بنا:

```
data = data.drop("Date", axis=1)
```

دعونا نلقي نظرة على الحالات المؤكدة اليومية لـ Covid-19:

```
import plotly.express as px
fig = px.bar(data, x='Date_YMD', y='Daily Confirmed')
fig.show()
```

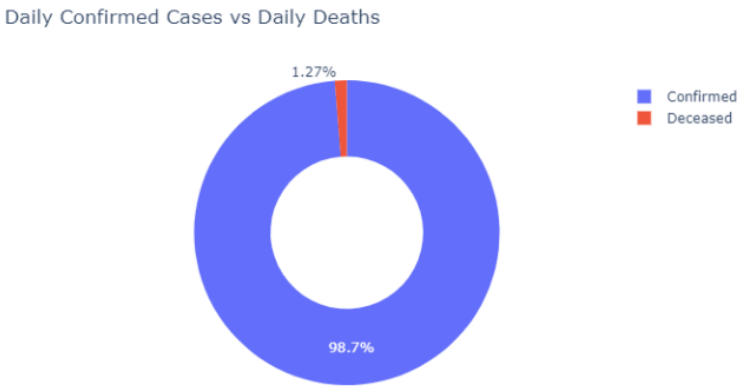


في تصوير البيانات أعلاه ، يمكننا أن نرى موجة عالية من حالات كوفيد ـ19 بين أبريل 2021 ومايو 2021.

تحليل معدل الوفيات Covid-19

الآن دعونا نصور معدل الوفيات (death rate) بسبب Covid-19:

```
cases = data["Daily Confirmed"].sum()
deceased = data["Daily Deceased"].sum()

labels = ["Confirmed", "Deceased"]
values = [cases, deceased]

fig = px.pie(data, values=values,
     names=labels,
     title='Daily Confirmed Cases vs Daily Deaths', hole=0.5)
fig.show()
```



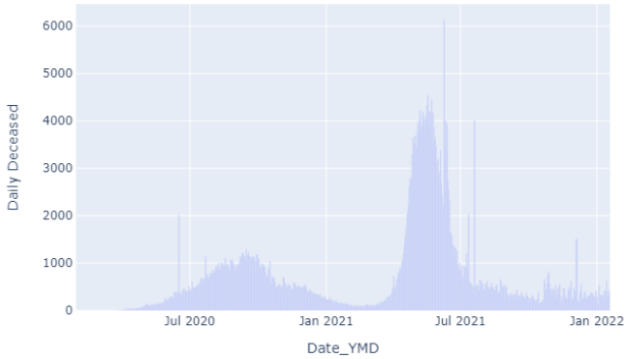
دعونا نحسب معدل الوفيات لـ Covid-19:

```
death_rate = (data["Daily Deceased"].sum() / data["Daily
Confirmed"].sum()) * 100
print(death_rate)
```

```
1.2840580507834722
```

الآن دعونا نلقي نظرة على الوفيات اليومية (daily deaths) لمرض Covid-19 :

```
import plotly.express as px
fig = px.bar(data, x='Date_YMD', y='Daily Deceased')
fig.show()
```



يمكننا أن نرى عددًا كبيرًا من الوفيات خلال الموجة العالية من حالات الإصابة بـcovid-19 .

نموذج التنبؤ بوفيات Covid-19

الآن دعنا ننتقل إلى مهمة التنبؤ بوفيات Covid-19 للأيام الثلاثين القادمة. سأستخدم هنا مكتبة AutoTS ، والتي تعد واحدة من أفضل مكتبات التعلم الآلي التلقائية لتحليل السلاسل الزمنية. إذا لم تكن قد استخدمت هذه المكتبة من قبل، فيمكنك تثبيتها عن طريق تنفيذ الأمر المذكور أدناهفي موجه الأوامر أو التيرمينال:

```
pip install autots
```

إليك الآن كيفية توقع وفيات Covid-19 باستخدام التعلم الآلي للأيام الثلاثين القادمة:

```
from autots import AutoTS
model = AutoTS(forecast_length=30, frequency='infer',
ensemble='simple')
model = model.fit(data, date_col="Date_YMD", value_col='Daily
Deceased', id_col=None)
prediction = model.predict()
forecast = prediction.forecast
print(forecast)
```

	Daily Deceased
2022-01-19	271.950000
2022-01-20	310.179787
2022-01-21	297.500000
2022-01-22	310.179787
2022-01-23	271.950000
2022-01-24	258.518302
2022-01-25	340.355520
2022-01-26	296.561343
2022-01-27	296.561343
2022-01-28	284.438262
2022-01-29	323.400000
2022-01-30	271.950000
2022-01-31	245.750000
2022-02-01	284.438262
2022-02-02	258.518302
2022-02-03	239.969607
2022-02-04	271.950000
2022-02-05	334.118953
2022-02-06	323.400000
2022-02-07	271.950000
2022-02-08	284.438262
2022-02-09	323.400000
2022-02-10	258.518302
2022-02-11	245.750000
2022-02-12	245.750000
2022-02-13	326.442185
2022-02-14	323.400000
2022-02-15	394.343619
2022-02-16	228.117431
2022-02-17	358.200000

الملخص

هذه هي الطريقة التي يمكننا بها توقع وفيات Covid-19 باستخدام التعلم الآلي باستخدام لغة برمجة بايثون. يمكننا استخدام البيانات التاريخية لحالات Covid-19 والوفيات للتنبؤ بعدد الوفياتفي المستقبل. يمكنك تنفيذ نفس الطريقة للتنبؤ بوفيات وموجات Covid-19 على أحدث مجموعة بيانات. آمل أن تكون قد أحببت هذا المقال عن التنبؤ بوفيات Covid-19 باستخدام التعلم الآلي.

16) التنبؤ بسعر السهم مع LSTM Stock Price Prediction with LSTM

LSTM تعني شبكات الذاكرة طويلة قصيرة المدى (Long Short-Term Memory Networks). إنها نوع من الشبكات العصبية المتكررة (recurrent neural network) التي تُستخدم بشكل شائع للتنبؤ بالانحدار (regression) والتنبؤ بالسلاسل الزمنية (time series) في التعلم الآلي. يمكنه حفظ البيانات لفترات طويلة، مما يميز الشبكات العصبية LSTM عن الشبكات العصبية الأخرى. إذا كنت تريد معرفة كيفية التنبؤ بأسعار الأسهم باستخدام LSTM، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة التنبؤ بأسعار الأسهم باستخدام LSTM باستخدام بايثون.

التنبؤ بسعر السهم مع LSTM

يعد استخدام LSTM أحد أفضل أساليب التعلم الآلي للتنبؤ بالسلسلة الزمنية. LSTMs هي شبكات عصبية متكررة مصممة لتذكر البيانات لفترة أطول. لذلك، كلما كنت تعمل على مشكلة حيث فشلت شبكتك العصبيةفي حفظ البيانات، يمكنك استخدام الشبكة العصبية LSTM. يمكنك قراءة المزيد عن LSTM هنا.

الآنفي هذا القسم، سوف آخذك خلال مهمة توقع أسعار الأسهم باستخدام LSTM باستخدام لغة برمجة بايثون. سأبدأ هذه المهمة عن طريق استيراد جميع مكتبات بايثون الضرورية وجمع أحدث بيانات أسعار أسهم Apple:

```
import pandas as pd
import yfinance as yf
import datetime
from datetime import date, timedelta
today = date.today()

d1 = today.strftime("%Y-%m-%d")
end_date = d1
d2 = date.today() - timedelta(days=5000)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2

data = yf.download('AAPL',
                   start=start_date ,
                   end=end_date ,
                   progress=False)
data["Date"] = data.index
data = data[["Date", "Open", "High", "Low", "Close ,"
"             Adj Close", "Volume"]]
data.reset_index(drop=True, inplace=True)
data.tail()
```

```
          Date        Open        High  ...       Close   Adj Close    Volume
3441 2021-12-27  177.089996  180.419998  ...  180.330002  180.330002  74919600
3442 2021-12-28  180.160004  181.330002  ...  179.289993  179.289993  79144300
3443 2021-12-29  179.330002  180.630005  ...  179.380005  179.380005  62348900
3444 2021-12-30  179.470001  180.570007  ...  178.199997  178.199997  59773000
3445 2021-12-31  178.089996  179.229996  ...  177.570007  177.570007  64025500

[5 rows x 7 columns]
```

يعطي مخطط الشموع اليابانية (candlestick chart) صورة واضحة عن الزيادة والانخفاض في أسعار الأسهم، لذلك دعونا نتخيل مخطط الشموع للبيانات قبل المضي قدمًا:

```
import plotly.graph_objects as go
figure = go.Figure(data=[go.Candlestick(x=data["Date’],
                                        open=data["Open"] ,
                                        high=data["High"],
                                        low=data["Low"] ,
                                        close=data["Close"])])
figure.update_layout(title = "Apple Stock Price Analysis”,
                     xaxis_rangeslider_visible=False)
figure.show()
```

دعنا الآن نلقي نظرة على ارتباط (correlation) جميع الأعمدة بعمود الإغلاق لأنه العمود الهدف:

```
correlation = data.corr()
print(correlation["Close"].sort_values(ascending=False))
```

```
Close        1.000000
Low          0.999890
High         0.999887
Adj Close    0.999845
Open         0.999783
Volume      -0.496325
Name: Close, dtype: float64
```

تدريب LSTM للتنبؤ بسعر السهم

سأبدأ الآن بتدريب نموذج LSTM للتنبؤ بأسعار الأسهم. سأقسم البيانات أولاً إلى مجموعات تدريب واختبار:

```
x = data[["Open", "High", "Low", "Volume"]]
y = data["Close"]
x = x.to_numpy()
y = y.to_numpy()
y = y.reshape(-1, 1)

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.2, random_state=42)
```

الآن سأقوم بإعداد هيكل شبكة عصبية لـLSTM :

```
from keras.models import Sequential
from keras.layers import Dense, LSTM
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape=
(xtrain.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_12 (LSTM)              (None, 4, 128)            66560

 lstm_13 (LSTM)              (None, 64)                49408

 dense_12 (Dense)            (None, 25)                1625

 dense_13 (Dense)            (None, 1)                 26

=================================================================
Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0
_________________________________________________________________
```

الآن إليك كيف يمكننا تدريب نموذج شبكتنا العصبية للتنبؤ بأسعار الأسهم:

```
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(xtrain, ytrain, batch_size=1, epochs=30)
```

```
Epoch 1/30
2756/2756 [==============================] - 18s 5ms/step - loss: 6.7984
Epoch 2/30
2756/2756 [==============================] - 15s 5ms/step - loss: 4.4978
Epoch 3/30
2756/2756 [==============================] - 15s 5ms/step - loss: 5.6511
Epoch 4/30
2756/2756 [==============================] - 15s 5ms/step - loss: 6.8347
Epoch 5/30
2756/2756 [==============================] - 15s 5ms/step - loss: 9.5083
Epoch 6/30
2756/2756 [==============================] - 15s 5ms/step - loss: 7.4367
Epoch 7/30
2756/2756 [==============================] - 15s 5ms/step - loss: 4.3043
Epoch 8/30
2756/2756 [==============================] - 15s 6ms/step - loss: 4.2213
Epoch 9/30
2756/2756 [==============================] - 15s 5ms/step - loss: 5.7352
Epoch 10/30
2756/2756 [==============================] - 15s 6ms/step - loss: 5.2137
Epoch 11/30
2756/2756 [==============================] - 15s 5ms/step - loss: 6.0945
Epoch 12/30
2756/2756 [==============================] - 14s 5ms/step - loss: 4.1032
Epoch 13/30
2756/2756 [==============================] - 15s 5ms/step - loss: 4.3637
Epoch 14/30
2756/2756 [==============================] - 15s 5ms/step - loss: 6.2240
Epoch 15/30
2756/2756 [==============================] - 15s 5ms/step - loss: 1.9857
```

```
Epoch 16/30
2756/2756 [==============================] - 15s 5ms/step - loss: 6.3982
Epoch 17/30
2756/2756 [==============================] - 15s 5ms/step - loss: 3.3015
Epoch 18/30
2756/2756 [==============================] - 15s 5ms/step - loss: 3.9104
Epoch 19/30
2756/2756 [==============================] - 15s 5ms/step - loss: 4.6564
Epoch 20/30
2756/2756 [==============================] - 15s 5ms/step - loss: 3.3215
Epoch 21/30
2756/2756 [==============================] - 15s 6ms/step - loss: 4.3116
Epoch 22/30
2756/2756 [==============================] - 15s 5ms/step - loss: 2.8147
Epoch 23/30
2756/2756 [==============================] - 15s 5ms/step - loss: 5.7586
Epoch 24/30
2756/2756 [==============================] - 16s 6ms/step - loss: 4.1890
Epoch 25/30
2756/2756 [==============================] - 17s 6ms/step - loss: 3.6991
Epoch 26/30
2756/2756 [==============================] - 15s 6ms/step - loss: 4.0951
Epoch 27/30
2756/2756 [==============================] - 15s 5ms/step - loss: 3.5940
Epoch 28/30
2756/2756 [==============================] - 16s 6ms/step - loss: 3.7180
Epoch 29/30
2756/2756 [==============================] - 15s 6ms/step - loss: 3.5864
Epoch 30/30
2756/2756 [==============================] - 15s 5ms/step - loss: 3.7422
<keras.callbacks.History at 0x7f8c37686790>
```

الآن دعنا نختبر هذا النموذج من خلال إعطاء قيم الإدخال وفقًا للميزات التي استخدمناها لتدريب هذا النموذج والتنبؤ بالنتيجة النهائية:

```
import numpy as np
#features = [Open, High, Low, Adj Close, Volume]
features = np.array ,177.070007 ,180.419998 ,177.089996]])
74919600 ]])
model.predict(features)
```

```
array([[179.95299]], dtype=float32)
```

إذن هذه هي الطريقة التي يمكننا بها استخدام هيكل الشبكة العصبية LSTM لمهمة التنبؤ بسعر السهم.

الملخص

LSTM تعني شبكات الذاكرة طويلة قصيرة المدى. إنها شبكة عصبية متكررة مصممة لتذكر البيانات لفترة أطول. يعد استخدام LSTM أحد أفضل أساليب التعلم الآلي للتنبؤ بالسلسلة الزمنية. آمل أن تكون قد أحببت هذه المقالة حول توقع أسعار الأسهم باستخدام LSTM باستخدام بايثون.

17) نظام توصية المقالات مع التعلم الآلي Article Recommendation System with Machine Learning

نظام التوصية (recommendation system) هو أحد تطبيقات علم البيانات التي يستخدمها أي تطبيق أو موقع ويب تقريبًا اليوم. تستخدم العديد من مواقع الويب اليوم نظام توصية للتوصية بالمقالات لقرائها. على سبيل المثال، يستخدم موقع Medium.com وحتى موقع الويب الذي تقرأ عليه هذه المقالة حاليًا نظامًا للتوصية بالتوصية بالمقالات لقرائه. لذلك، إذا كنت ترغب في معرفة كيفية إنشاء نظام توصية بالمقالات باستخدام التعلم الآلي، فهذه المقالة مناسبة لك. في هذه المقالة، سوف أطلعك على كيفية إنشاء نظام توصيات المقالة باستخدام التعلم الآلي باستخدام بايثون.

نظام التوصية بالمقالات

هناك العديد من الطرق لإنشاء أنظمة التوصية. لإنشاء نظام توصية بالمقالات، نحتاج إلى التركيز على المحتوى (content) بدلاً من اهتمام المستخدم. على سبيل المثال، إذا قرأ المستخدم مقالة تستند إلى المجموعات، فيجب أن تستند جميع المقالات الموصى بها أيضًا إلى المجموعات. لذلك للتوصية بالمقالات بناءً على المحتوى:

1. نحن بحاجة إلى فهم محتوى المقال.
2. تطابق المحتوى مع جميع المقالات الأخرى.
3. والتوصية بالمقالات الأكثر ملاءمة للمقالة التي يقرأها القارئ بالفعل.

لهذه المهمة، يمكننا استخدام مفهوم تشابه جيب التمام (cosine similarity) في التعلم الآلي. تشابه جيب التمام هو طريقة لبناء أنظمة التوصية بناءً على المحتوى. يتم استخدامه لإيجاد أوجه التشابه بين قطعتين مختلفتين من المستندات النصية. لذلك يمكننا استخدام تشابه جيب التمام لبناء نظام توصية بالمقالات. في القسم أدناه، سوف أطلعك على كيفية إنشاء نظام توصية بالمقالات باستخدام التعلم الآلي باستخدام بايثون.

نظام التوصية بالمقالات باستخدام بايثون

لإنشاء نظام توصية بالمقالات، قمت بجمع بيانات حول بعض المقالات على هذا الموقع نفسه. لذلك دعونا نستورد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لإنشاء نظام توصية بالمقالات:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction import text
from sklearn.metrics.pairwise import cosine_similarity
```

```
data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/articles.csv", encoding='latin1')
data.head()
```

	Article	Title
0	Data analysis is the process of inspecting and...	Best Books to Learn Data Analysis
1	The performance of a machine learning algorith...	Assumptions of Machine Learning Algorithms
2	You must have seen the news divided into categ...	News Classification with Machine Learning
3	When there are only two classes in a classific...	Multiclass Classification Algorithms in Machin...
4	The Multinomial Naive Bayes is one of the vari...	Multinomial Naive Bayes in Machine Learning

مجموعة البيانات هذه جاهزة تمامًا للاستخدام لإنشاء نظام توصية، لذلك دعونا نستخدم خوارزمية تشابه جيب التمام ونكتب دالة بايثون للتوصية بالمقالات:

```
articles = data["Article"].tolist()
uni_tfidf = text.TfidfVectorizer(input=articles,
stop_words="english")
uni_matrix = uni_tfidf.fit_transform(articles)
uni_sim = cosine_similarity(uni_matrix)
def recommend_articles(x):
    return ", ".join(data["Title"].loc[x.argsort()[-5:-1]])
data["Recommended Articles"] = [recommend_articles(x) for x in uni_sim]
data.head()
```

	Article	Title	Recommended Articles
0	Data analysis is the process of inspecting and...	Best Books to Learn Data Analysis	Introduction to Recommendation Systems, Best B...
1	The performance of a machine learning algorith...	Assumptions of Machine Learning Algorithms	Applications of Deep Learning, Best Books to L...
2	You must have seen the news divided into categ...	News Classification with Machine Learning	Language Detection with Machine Learning, Appl...
3	When there are only two classes in a classific...	Multiclass Classification Algorithms in Machin...	Assumptions of Machine Learning Algorithms, Be...
4	The Multinomial Naive Bayes is one of the vari...	Multinomial Naive Bayes in Machine Learning	Assumptions of Machine Learning Algorithms, Me...

كما ترى من الإخراج أعلاه، تمت إضافة عمود جديد إلى مجموعة البيانات التي تحتوي على عناوين جميع المقالات الموصى بها. دعنا الآن نرى جميع التوصيات لمقالة:

```
print(data["Recommended Articles"][22])
```

```
BIRCH Clustering in Machine Learning, Clustering Algorithms in Machine Learning, DBSCAN Clustering in Machine Learning, K-Means Clustering in Machine Learning
```

يحتوي الفهرس 22 على مقال عن التجميع المتكتل (agglomerated clustering)، وتستند جميع المقالات الموصى بها أيضًا إلى مفاهيم التجميع، لذلك يمكننا القول إن نظام التوصية هذا يمكن أيضًا أن يعطي نتائج رائعةفي الوقت الفعلي.

الملخص

هذه هي الطريقة التي يمكنك بها بناء نظام توصية بالمقالات. تستخدم الكثير من مواقع الويب اليوم نظام توصية للتوصية بالمقالات لقرائها. آمل أن تكون قد أحببت هذه المقالة حول كيفية إنشاء نظام توصية بالمقالات باستخدام التعلم الآلي باستخدام بايثون.

18) التنبؤ بالمبيعات المستقبلية مع التعلم الآلي Future Sales Prediction with Machine Learning

يساعد التنبؤ بالمبيعات المستقبلية (Future Sales Prediction) لمنتج ما الشركةفي إدارة تكلفة التصنيع والإعلان للمنتج. هناك العديد من الفوائد للتنبؤ بالمبيعات المستقبلية للمنتج. لذلك إذا كنت تريد أن تتعلم التنبؤ بالمبيعات المستقبلية لمنتج ما باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بالمبيعات المستقبلية من خلال التعلم الآلي باستخدام بايثون.

التنبؤ بالمبيعات المستقبلية (دراسة حالة)

تحتوي مجموعة البيانات الواردة هنا على بيانات حول مبيعات المنتج. تتعلق مجموعة البيانات بتكلفة الإعلان التي يتكبدها النشاط التجاري على منصات إعلانية مختلفة. فيما يلي وصف لجميع الأعمدةفي مجموعة البيانات:

1. **TV**: تكلفة الإعلان بالدولار للإعلان على التلفزيون؛
2. **Radio**: تكلفة الإعلان بالدولار للإعلان على الراديو؛
3. **Newspaper**: تكلفة الإعلان التي تنفق بالدولار للإعلانفي الجريدة؛
4. **Sales**: عدد الوحدات المباعة؛

لذلك،في مجموعة البيانات أعلاه، تعتمد مبيعات المنتج على تكلفة الإعلان عن المنتج. آمل أن تكون قد فهمت الآن كل شيء عن مجموعة البيانات هذه. الآنفي القسم أدناه، سوف آخذك خلال مهمة التنبؤ بالمبيعات المستقبلية باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بالمبيعات المستقبلية باستخدام بايثون

لنبدأ مهمة التنبؤ بالمبيعات المستقبلية باستخدام التعلم الآلي عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/advertising.csv")
print(data.head())
```

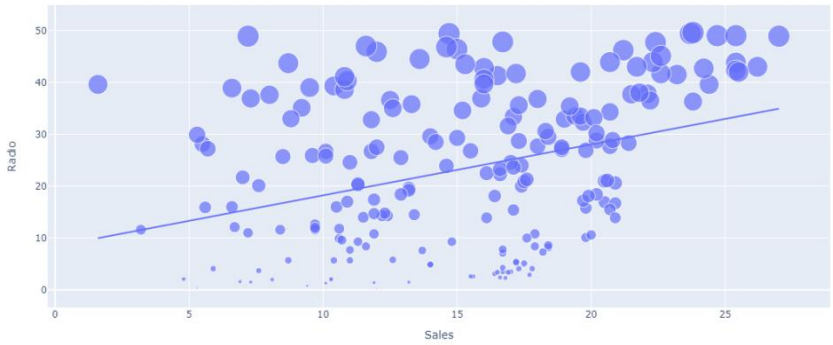
دعونا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

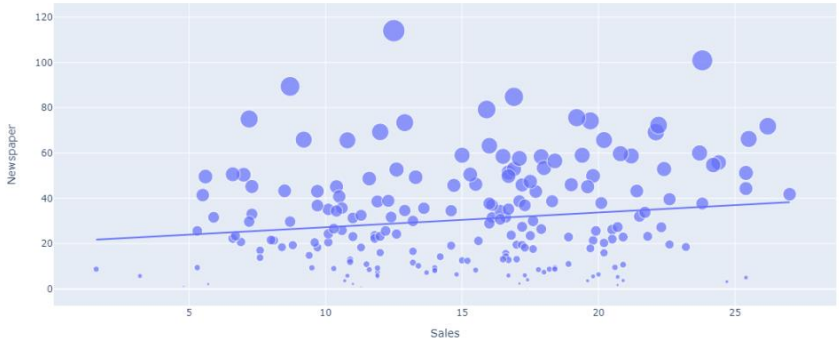
لذلك لا تحتوي مجموعة البيانات هذه على أي قيم فارغة. دعنا الآن نرسم العلاقة بين المبلغ الذي يتم إنفاقه على الإعلان على التلفزيون والوحدات المباعة (units sold):

```
figure = px.scatter(data_frame = data, x="Sales ",
                    y="Newspaper", size="Newspaper",
trendline="ols")
figure.show()
```



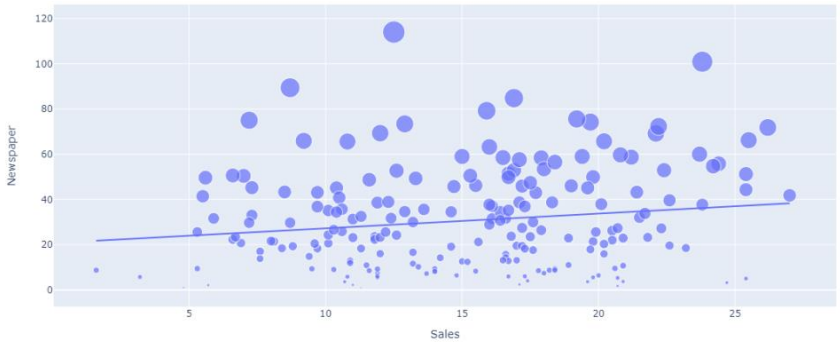
دعنا الآن نرسم العلاقة بين المبلغ الذي يتم إنفاقه على الإعلان على الراديو والوحدات المباعة:

```
figure = px.scatter(data_frame = data, x="Sales ",
        y="Radio", size="Radio", trendline="ols ")
figure.show()
```



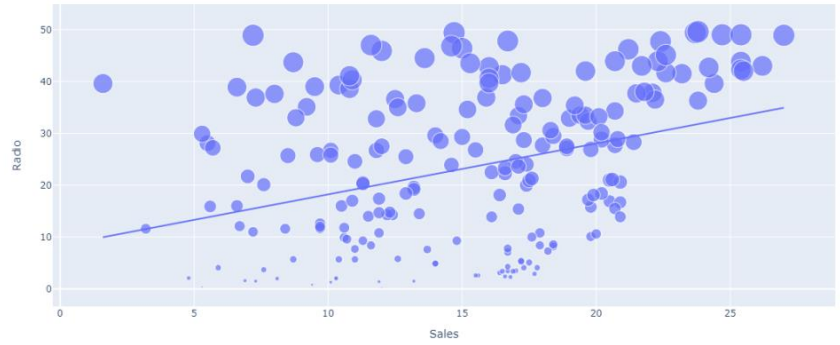
دعنا الآن نرسم العلاقة بين المبلغ الذي يتم إنفاقه على الإعلانفي الصحف والوحدات المباعة:

```
figure = px.scatter(data_frame = data, x="Sales ",
        y="Newspaper", size="Newspaper", trendline="ols ")
figure.show()
```



دعنا الآن نرسم العلاقة بين المبلغ الذي يتم إنفاقه على الإعلان على الراديو والوحدات المباعة:

```
figure = px.scatter(data_frame = data, x="Sales",
        y="Radio", size="Radio", trendline="ols")
figure.show()
```



من بين كل المبلغ الذي يتم إنفاقه على الإعلان على منصات مختلفة، أستطيع أن أرى أن المبلغ الذي يتم إنفاقه على الإعلان عن المنتج على التلفزيون يؤدي إلى زيادة مبيعات المنتج. دعنا الآن نلقي نظرة على ارتباط جميع الأعمدة بعمود المبيعات:

```
correlation = data.corr()
print(correlation["Sales"].sort_values(ascending=False))
```

```
Sales        1.000000
TV           0.901208
Radio        0.349631
Newspaper    0.157960
Name: Sales, dtype: float64
```

نموذج التنبؤ بالمبيعات المستقبلية

الآنفي هذا القسم، سأقوم بتدريب نموذج التعلم الآلي للتنبؤ بالمبيعات المستقبلية للمنتج. ولكن قبل تدريب النموذج، دعنا نقسم البيانات إلى مجموعات تدريب واختبار:

```
x = np.array(data.drop(["Sales"], 1))
y = np.array(data["Sales"])
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                    test_size=0.2,
                    random_state=42)
```

دعنا الآن ندرب النموذج للتنبؤ بالمبيعات المستقبلية:

```
model = LinearRegression()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

دعنا الآن قيم الإدخالفي النموذج وفقًا للميزات التي استخدمناها لتدريبه والتنبؤ بعدد وحدات المنتج التي يمكن بيعها بناءً على المبلغ الذي يتم إنفاقه على إعلاناته على منصات مختلفة:

```
#features = [[TV, Radio, Newspaper]]
features = np.array([[69.2,37.8,230.1]])
print(model.predict(features))
```

```
[21.37254028]
```

الملخص

هذه هي الطريقة التي يمكننا بها تدريب نموذج التعلم الآلي للتنبؤ بالمبيعات المستقبلية للمنتج. يساعد توقع المبيعات المستقبلية لمنتج ما الشركةفي إدارة تكلفة التصنيع والإعلان للمنتج. آمل أن تكون قد أحببت هذه المقالة حول تنبؤات المبيعات المستقبلية باستخدام التعلم الآلي.

19) التنبؤ بسعر سهم Netflix مع التعلم الآلي Netflix Stock Price Prediction with Machine Learning

Netflix هي واحدة من أكثر منصات بث خدمة وسائط فائقة (OTT) شيوعًا. تقدم مجموعة كبيرة من المسلسلات التلفزيونية والأفلام وتمتلك منتجاتها المعروفة باسم Netflix Originals. دائمًا ما يراقب الأشخاص الناشطون للغايةفي استثمارات سوق الأوراق المالية شركات مثل Netflix نظرًا لشعبيتها. إذا كنت أحدهم ممن يرغبونفي التعرف على كيفية التنبؤ بسعر أسهم Netflix باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة توقع أسعار أسهم Netflix باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بسعر سهم Netflix مع التعلم الآلي

للتنبؤ بأسعار أسهم Netflix مع التعلم الآلي، سأستخدم الشبكة العصبية LSTM لأنها واحدة من أفضل الطرق لتحليل الانحدار والتنبؤ بالسلسلة الزمنية. لذلك هنا، سأبدأ باستيراد مكتبات بايثون الضرورية وجمع أحدث بيانات أسعار أسهم Netflix:

```
import pandas as pd
import yfinance as yf
import datetime
from datetime import date, timedelta
today = date.today()

d1 = today.strftime("%Y-%m-%d")
end_date = d1
d2 = date.today() - timedelta(days=5000)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2

data = yf.download('NFLX,'
          start=start_date,
          end=end_date,
          progress=False)
data["Date"] = data.index
data = data[["Date", "Open", "High", "Low", "Close", "Adj
Close", "Volume"]]
data.reset_index(drop=True, inplace=True)
print(data.tail())
```

	Date	Open	High	...	Close	Adj Close	Volume
3442	2022-02-01	432.959991	458.480011	...	457.130005	457.130005	22568100
3443	2022-02-02	448.250000	451.980011	...	429.480011	429.480011	14346000
3444	2022-02-03	421.440002	429.260010	...	405.600006	405.600006	9905200
3445	2022-02-04	407.309998	412.769989	...	410.170013	410.170013	7782400
3446	2022-02-07	410.170013	412.350006	...	402.100006	402.100006	8228000

[5 rows x 7 columns]

يتم جمع مجموعة البيانات أعلاه باستخدام yfinance API في بايثون. يمكنك معرفة المزيد عنها من هنا. الآن دعنا نرسم بيانات أسعار الأسهم في Netflix باستخدام مخطط الشموع (candlestick chart) لأنه يعطي صورة واضحة عن الزيادة والنقصان في أسعار الأسهم:

```
import plotly.graph_objects as go
figure = go.Figure(data=[go.Candlestick(x=data["Date,["
                                        open=data["Open"] ,
                                        high=data["High"],
                                        low=data["Low"] ,
                                        close=data["Close"])])
figure.update_layout(title = "Netflix Stock Price Analysis",
                     xaxis_rangeslider_visible=False)
figure.show()
```

دعنا الآن نلقي نظرة على ارتباط (correlation) جميع الأعمدة بعمود الإغلاق (Close column):

```
correlation = data.corr()
print(correlation["Close"].sort_values(ascending=False))
```

```
Close        1.000000
Adj Close    1.000000
High         0.999837
Low          0.999831
Open         0.999643
Volume      -0.395022
Name: Close, dtype: float64
```

تدريب LSTM على التنبؤ بسعر سهم Netflix

سأقوم الآن بتدريب نموذج الشبكة العصبية LSTM لمهمة توقع أسعار سهم Netflix باستخدام بايثون. سأقوم هنا أولاً بتقسيم البيانات إلى مجموعات تدريب واختبار:

```
x = data[["Open", "High", "Low", "Volume"]]
y = data["Close"]
x = x.to_numpy()
y = y.to_numpy()
```

```
y = y.reshape(1,-1)

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                    test_size=0.2,
                     random_state=42)
```

الآن سأقوم بإعداد هيكل الشبكة العصبيةLSTM :

```
from keras.models import Sequential
from keras.layers import Dense, LSTM
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape=
(xtrain.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 4, 128)            66560

 lstm_1 (LSTM)               (None, 64)                49408

 dense (Dense)               (None, 25)                1625

 dense_1 (Dense)             (None, 1)                 26

=================================================================
Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0
_________________________________________________________________
```

الآن إليك كيف يمكننا تدريب نموذج شبكة عصبية LSTM للتنبؤ بأسعار أسهم Netflix باستخدام بايثون:

```
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(xtrain, ytrain, batch_size=1, epochs=30)
```

```
Epoch 1/30
2757/2757 [==============================] - 26s 8ms/step - loss: 7604.0146
Epoch 2/30
2757/2757 [==============================] - 21s 8ms/step - loss: 631.5478
Epoch 3/30
2757/2757 [==============================] - 21s 8ms/step - loss: 464.4248
Epoch 4/30
2757/2757 [==============================] - 21s 8ms/step - loss: 372.2728
Epoch 5/30
2757/2757 [==============================] - 23s 8ms/step - loss: 393.3465
```

```
Epoch 6/30
2757/2757 [==============================] - 23s 8ms/step - loss: 388.6576
Epoch 7/30
2757/2757 [==============================] - 24s 9ms/step - loss: 256.3762
Epoch 8/30
2757/2757 [==============================] - 24s 9ms/step - loss: 233.4001
Epoch 9/30
2757/2757 [==============================] - 24s 9ms/step - loss: 277.1848
Epoch 10/30
2757/2757 [==============================] - 24s 9ms/step - loss: 271.6196
Epoch 11/30
2757/2757 [==============================] - 21s 8ms/step - loss: 314.4778
Epoch 12/30
2757/2757 [==============================] - 22s 8ms/step - loss: 310.0940
Epoch 13/30
2757/2757 [==============================] - 22s 8ms/step - loss: 230.9971
Epoch 14/30
2757/2757 [==============================] - 22s 8ms/step - loss: 320.0016
Epoch 15/30
2757/2757 [==============================] - 21s 8ms/step - loss: 266.8751
Epoch 16/30
2757/2757 [==============================] - 22s 8ms/step - loss: 184.3275
Epoch 17/30
2757/2757 [==============================] - 23s 8ms/step - loss: 187.5371
Epoch 18/30
2757/2757 [==============================] - 23s 8ms/step - loss: 262.0464
Epoch 19/30
2757/2757 [==============================] - 23s 9ms/step - loss: 124.5728
Epoch 20/30
2757/2757 [==============================] - 22s 8ms/step - loss: 235.3674
Epoch 21/30
2757/2757 [==============================] - 22s 8ms/step - loss: 210.1229
Epoch 22/30
2757/2757 [==============================] - 22s 8ms/step - loss: 149.8269
Epoch 23/30
2757/2757 [==============================] - 22s 8ms/step - loss: 137.0508
Epoch 24/30
2757/2757 [==============================] - 22s 8ms/step - loss: 182.6988
Epoch 25/30
2757/2757 [==============================] - 22s 8ms/step - loss: 139.5222
Epoch 26/30
2757/2757 [==============================] - 23s 8ms/step - loss: 174.5475
Epoch 27/30
2757/2757 [==============================] - 23s 9ms/step - loss: 139.0385
Epoch 28/30
2757/2757 [==============================] - 23s 8ms/step - loss: 135.3190
Epoch 29/30
2757/2757 [==============================] - 21s 8ms/step - loss: 122.6136
Epoch 30/30
2757/2757 [==============================] - 21s 8ms/step - loss: 182.0294
<keras.callbacks.History at 0x7f0057f21790>
```

الآن دعنا نختبر النموذج من خلال إعطاء المدخلات وفقًا للميزات التي استخدمناها لتدريب هذا النموذج للتنبؤ بالنتائج النهائية :

```
import numpy as np
#features = [Open, High, Low, Adj Close, Volume]
```

```
features = np.array([[,398.200012,427.700012,401.970001
20047500]])
model.predict(features)
```

```
array([[408.57364]], dtype=float32)
```

هذه هي الطريقة التي يمكننا بها تدريب نموذج شبكة عصبية LSTM لمهمة للتنبؤ بأسعار أسهم Netflixمع التعلم الآلي باستخدام بايثون.

الملخص

هذه هي الطريقة التي يمكننا بها استخدام التعلم الآلي للتنبؤ بأسعار أسهم Netflix. Netflix هي واحدة من أكثر منصات بث OTT شيوعًا. دائمًا ما يراقب الأشخاص الناشطون للغايةفي استثمارات سوق الأوراق المالية شركات مثل Netflix نظرًا لشعبيتها. آمل أن تكون قد أحببت هذه المقالة حول مهمة توقع أسعار أسهم Netflix مع التعلم الآلي باستخدام بايثون.

20) اكتشاف الاجهاد باستخدام التعلم الآلي Stress Detection with Machine Learning

يهدد الاجهاد (Stress) والقلق (anxiety) والاكتئاب (depression) الصحة العقلية (mental health) للناس. كل شخص لديه سبب لعيش حياة مرهقة. غالبًا ما يشارك الأشخاص مشاعرهم على منصات التواصل الاجتماعي مثل Instagram في شكل منشورات وقصص، وعلى Reddit في شكل طلب اقتراحات حول حياتهم على subreddits. في السنوات القليلة الماضية، تقدم العديد من منشئي المحتوى لإنشاء محتوى لمساعدة الأشخاص في صحتهم العقلية. يمكن للعديد من المنظمات استخدام الكشف عن الإجهاد للعثور على مستخدمي الوسائط الاجتماعية الذين يتم الضغط عليهم لمساعدتهم بسرعة. لذلك إذا كنت تريد معرفة كيفية استخدام التعلم الآلي لاكتشاف الضغط على منشورات وسائل التواصل الاجتماعي، فهذه المقالة مناسبة لك. في هذه المقالة، سوف آخذك خلال مهمة الكشف عن الاجهاد باستخدام التعلم الآلي باستخدام بايثون.

اكتشاف الاجهاد باستخدام التعلم الآلي

يعد اكتشاف الاجهاد مهمة صعبة، حيث يوجد الكثير من الكلمات التي يمكن أن يستخدمها الأشخاص في مشاركاتهم والتي يمكن أن توضح ما إذا كان الشخص يعاني من ضغوط نفسية أم لا. أثناء البحث عن مجموعات البيانات التي يمكنني استخدامها لتدريب نموذج التعلم الآلي للكشف عن الإجهاد، وجدت مجموعة بيانات على Kaggle تحتوي على 116 عمودًا. نحتاج فقط إلى استخدام عمود النص (text) والتسمية (label) لهذه المهمة.

تحتوي مجموعة البيانات التي أستخدمها لهذه المهمة على بيانات منشورة على subreddits تتعلق بالصحة العقلية. تحتوي مجموعة البيانات هذه على العديد من مشكلات الصحة العقلية التي يشاركها الأشخاص في حياتهم. لحسن الحظ، تم تصنيف مجموعة البيانات هذه على أنها 0 و 1، حيث يشير 0 إلى عدم وجود إجهاد و 1 يشير إلى الإجهاد. لذلك في القسم أدناه، سوف آخذك خلال مهمة الكشف عن التوتر في منشورات وسائل التواصل الاجتماعي باستخدام بايثون.

اكتشاف الإجهاد باستخدام بايثون

لنبدأ الآن مهمة اكتشاف الإجهاد باستخدام التعلم الآلي. سأبدأ هذه المهمة عن طريق استيراد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لهذه المهمة:

```
import pandas as pd
import numpy as np
data = pd.read_csv("stress.csv")
print(data.head())
```

```
          subreddit post_id  ... syntax_fk_grade sentiment
0             ptsd  8601tu  ...        3.253573 -0.002742
1       assistance  81brx9  ...        8.828316  0.292857
2             ptsd  9ch1zh  ...        7.841667  0.011894
3    relationships  7rorpp  ...        4.104027  0.141671
4  survivorsofabuse  9p2gbc  ...        7.910952 -0.204167

[5 rows x 116 columns]
```

دعونا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

```
subreddit                  0
post_id                    0
sentence_range             0
text                       0
id                         0
                          ..
lex_dal_avg_pleasantness   0
social_upvote_ratio        0
social_num_comments        0
syntax_fk_grade            0
sentiment                  0
Length: 116, dtype: int64
```

لذلك لا تحتوي مجموعة البيانات هذه على أي قيم فارغة. دعنا الآن نجهز عمود النصي لمجموعة البيانات هذه لتنظيف عمود النص بكلمات التوقف (stopwords) والروابط والرموز الخاصة وأخطاء اللغة:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
```

```
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
   return text
data["text"]=data["text"].apply(clean)
```

دعنا الآن نلقي نظرة على الكلمات الأكثر استخدامًا من قبل الأشخاص الذين يشاركون حول مشاكل حياتهم على وسائل التواصل الاجتماعي من خلال تصوير سحابة الكلمات (word cloud)في عمود النص:

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
text = " ".join(i for i in data.text)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
         background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

نموذج اكتشاف الإجهاد

يحتوي عمود التسميةفي مجموعة البيانات هذه على تسميات مثل 0 و 1. 0 يعني عدم وجود اجهاد، ويعني 1 إجهاد. سأستخدم تسميات الإجهاد وعدم الإجهاد بدلاً من 1 و 0. لذا فلنعد هذا العمود وفقًا لذلك ونحدد أعمدة النص والتسمية لعملية تدريب نموذج التعلم الآلي:

```
data["label"] = data["label"].map({0: "No Stress", 1:
"Stress"})
data = data[["text", "label"]]
```

```
print(data.head())
```

```
                                        text      label
0  said felt way sugget go rest trigger ahead you...     Stress
1  hey rassist sure right place post goe  im curr...  No Stress
2  mom hit newspap shock would know dont like pla...     Stress
3  met new boyfriend amaz kind sweet good student...     Stress
4  octob domest violenc awar month domest violenc...     Stress
```

الآن سأقسم مجموعة البيانات هذه إلى مجموعات تدريب واختبار:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split

x = np.array(data["text"])
y = np.array(data["label"])

cv = CountVectorizer()
X = cv.fit_transform(x)
xtrain, xtest, ytrain, ytest = train_test_split(X, y,
                                 test_size=0.33,
                                 random_state=42)
```

نظرًا لأن هذه المهمة تستند إلى مشكلة التصنيف الثنائي (binary classification)، فسوف أستخدم خوارزمية Bernoulli Naive Bayes، والتي تعد واحدة من أفضل الخوارزميات لمشاكل التصنيف الثنائي. لذلك دعونا ندرب على نموذج الكشف عن الاجهاد:

```
from sklearn.naive_bayes import BernoulliNB
model = BernoulliNB()
model.fit(xtrain, ytrain)
```

دعنا الآن نختبر أداء نموذجنا على بعض الجمل العشوائية بناءً على الصحة العقلية:

```
user = input("Enter a Text: ")
data = cv.transform([user]).toarray()
output = model.predict(data)
print(output)
```

```
Enter a Text: People need to take care of their mental health
['No Stress']
```

```
user = input("Enter a Text: ")
data = cv.transform([user]).toarray()
output = model.predict(data)
print(output)
```

```
Enter a Text: Sometime I feel like I need some help
['Stress']
```

كما ترون، يمكننا أن نرى نتائج جيدة من نموذج التعلم الآلي الخاص بنا. هذه هي الطريقة التي يمكنك بها تدريب نموذج كشف الإجهاد لاكتشاف الاجهاد من منشورات وسائل التواصل الاجتماعي. يمكن تحسين نموذج التعلم الآلي هذا عن طريق تزويده بمزيد من البيانات.

الملخص

هذه هي الطريقة التي يمكنك بها تدريب نموذج التعلم الآلي لاكتشاف الإجهاد من منشورات وسائل التواصل الاجتماعي. غالبًا ما يشارك الناس مشاعرهم على منصات التواصل الاجتماعي. يمكن للعديد من المنظمات استخدام الكشف عن الإجهاد للعثور على مستخدمي الوسائط الاجتماعية الذين يتم الضغط عليهم لمساعدتهم بسرعة. آمل أن تكون قد أحببت هذه المقالة حول اكتشاف الإجهاد باستخدام التعلم الآلي باستخدام بايثون.

21) التنبؤ بالطلب على المنتج باستخدام التعلم الآلي Product Demand Prediction with Machine Learning

يجب أن تكون قد درست أن الطلب على المنتج (demand for a product) يختلف باختلاف سعره. إذا أخذت أمثلة من العالم الحقيقي، فسترى ما إذا كان المنتج ليس ضرورة، ثم يتناقص الطلب مع زيادة سعره ويزداد الطلب مع انخفاض سعره. إذا كنت تريد معرفة كيف يمكننا التنبؤ بالطلب على منتج باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة التنبؤ بالطلب على المنتج باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بالطلب على المنتج (دراسة حالة)

تخطط شركة منتج لتقديم خصومات على منتجاتها خلال موسم العطلات القادم. تريد الشركة العثور على السعر الذي يمكن أن يكون فيه منتجها صفقة أفضل مقارنة بمنافسيها. بالنسبة لهذه المهمة، قدمت الشركة مجموعة بيانات بالتغييرات السابقةفي المبيعات بناءً على تغيرات الأسعار. تحتاج إلى تدريب نموذج يمكنه التنبؤ بالطلب على المنتجفي السوق بشرائح أسعار مختلفة.

تحتوي مجموعة البيانات التي لدينا لهذه المهمة على بيانات حول:

1. معرّف المنتج (the product id)؛
2. معرف المتجر (store id).
3. السعر الإجمالي (total price) الذي تم بيع المنتج به؛
4. السعر الأساسي (base price) الذي تم بيع المنتج به؛
5. الوحدات المباعة (Units sold) (الكمية المطلوبة)؛

آمل أن تفهم الآن نوع بيانات المشكلة التي ستحصل عليها لمهمة التنبؤ بالطلب على المنتج.في القسم أدناه، سوف أطلعك على التنبؤ بالطلب على المنتج باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بالطلب على المنتج باستخدام بايثون

لنبدأ باستيراد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لمهمة التنبؤ بالطلب على المنتج:

```
import pandas as pd
```

```
import numpy as np
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/demand.csv")
data.head()
```

	ID	Store ID	Total Price	Base Price	Units Sold
0	1	8091	99.0375	111.8625	20
1	2	8091	99.0375	99.0375	28
2	3	8091	133.9500	133.9500	19
3	4	8091	133.9500	133.9500	44
4	5	8091	141.0750	141.0750	52

دعنا الآن نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

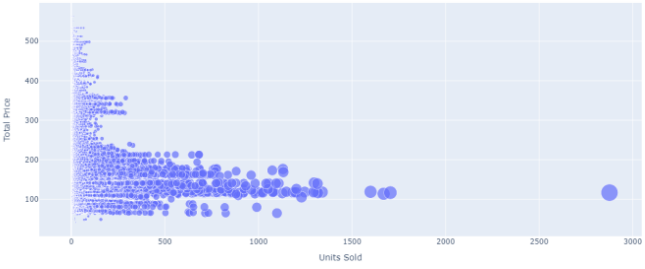
```
ID            0
Store ID      0
Total Price   1
Base Price    0
Units Sold    0
dtype: int64
```

لذلك، تحتوي مجموعة البيانات على قيمة واحدة مفقودة فقطفي عمود السعر الإجمالي (Total Price)، سأقوم بإزالة هذا الصف بالكامل في الوقت الحالي:

```
data = data.dropna()
```

دعونا الآن نحلل العلاقة بين السعر والطلب على المنتج. سأستخدم هنا مخطط مبعثر (scatter plot) لمعرفة كيف يختلف الطلب على المنتج مع تغير السعر:

```
fig = px.scatter(data, x="Units Sold", y="Total Price,"
       size='Units Sold ')
fig.show()
```

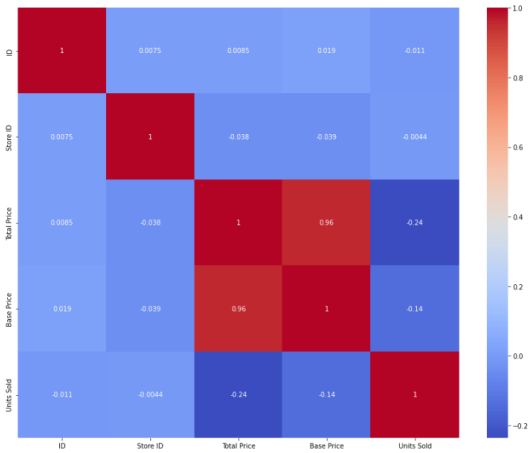


يمكننا أن نرى أن معظم نقاط البيانات تظهر أن مبيعات المنتج تتزايد حيث أن السعر يتناقص مع بعض الاستثناءات. دعنا الآن نلقي نظرة على الارتباط (correlation) بين ميزات مجموعة البيانات:

```
print(data.corr())
```

```
                ID  Store ID  Total Price  Base Price  Units Sold
ID        1.000000  0.007464     0.008473    0.018932   -0.010616
Store ID  0.007464  1.000000    -0.038315   -0.038848   -0.004372
Total Price  0.008473 -0.038315     1.000000    0.958885   -0.235625
Base Price   0.018932 -0.038848     0.958885    1.000000   -0.140032
Units Sold  -0.010616 -0.004372    -0.235625   -0.140032    1.000000
```

```
correlations = data.corr(method='pearson')
plt.figure(figsize=(15, 12))
sns.heatmap(correlations, cmap="coolwarm", annot=True)
plt.show()
```



نموذج التنبؤ بالطلب على المنتج

دعنا الآن ننتقل إلى مهمة تدريب نموذج التعلم الآلي للتنبؤ بالطلب على المنتج بأسعار مختلفة. سأختار عمود السعر الإجمالي (Total Price) والسعر الأساسي (Base Price) كميزات لتدريب النموذج، وعمود الوحدات المباعة (Units sold) كعناوين للنموذج:

```
x = data[["Total Price", "Base Price"]]
y = data["Units Sold"]
```

دعنا الآن نقسم البيانات إلى مجموعات تدريب واختبار ونستخدم خوارزمية انحدار شجرة القرار (decision tree regression) لتدريب نموذجنا:

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                    test_size=0.2,
                    random_state=42)
from sklearn.tree import DecisionTreeRegressor
```

```
model = DecisionTreeRegressor()
model.fit(xtrain, ytrain)
```

الآن دعنا ندخل الميزات (السعر الإجمالي، السعر الأساسي)في النموذج ونتوقع الكمية التي يمكن طلبها بناءً على هذه القيم:

```
#features = [["Total Price", "Base Price"]]
features = np.array([[140.00,133.00]])
model.predict(features)
```

```
array([27.])
```

الملخص

هذه هي الطريقة التي يمكنك بها تدريب نموذج التعلم الآلي لمهمة التنبؤ بالطلب على المنتج باستخدام بايثون. السعر هو أحد العوامل الرئيسية التي تؤثر على الطلب على المنتج. إذا لم يكن المنتج ضرورة، فإن قلة من الناس فقط هم من يشترون المنتج حتى لو ارتفع السعر. آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بالطلب على المنتج من خلال التعلم الآلي باستخدام بايثون.

22) التنبؤ بأسعار الكهرباء مع التعلم الآلي Electricity Price Prediction with Machine Learning

يعتمد سعر الكهرباء (Electricity Price) على عدة عوامل. يساعد التنبؤ بسعر الكهرباء العديد من الشركات على فهم مقدار الكهرباء التي يتعين عليهم دفعها كل عام. تعتمد مهمة التنبؤ بأسعار الكهرباء على دراسة حالة تحتاج فيها إلى توقع السعر اليومي للكهرباء بناءً على الاستهلاك اليومي للآلات الثقيلة التي تستخدمها الشركات. لذلك إذا كنت تريد معرفة كيفية التنبؤ بسعر الكهرباء، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة التنبؤ بأسعار الكهرباء مع التعلم الآلي باستخدام بايثون.

التنبؤ بأسعار الكهرباء (دراسة حالة)

افترض أن عملك يعتمد على خدمات الحوسبة حيث تختلف الطاقة التي تستهلكها أجهزتك على مدار اليوم. لا تعرف التكلفة الفعلية للكهرباء التي تستهلكها الآلات على مدار اليوم، لكن المنظمة زودتك ببيانات تاريخية عن سعر الكهرباء التي تستهلكها الآلات. فيما يلي معلومات البيانات المتوفرة لدينا لمهمة التنبؤ بأسعار الكهرباء:

1. DateTime: تاريخ ووقت السجل.
2. Holiday: يحتوي على اسم العطلة إذا كان اليوم عطلة وطنية.
3. HolidayFlag: يحتوي على 1 إذا كانت عطلة البنوك وإلا 0.
4. DayOfWeek: يحتوي على قيم بين 0ـ6 حيث يكون 0 هو يوم الإثنين.
5. WeekOfYear: أسبوع من السنة.
6. Day: يوم التاريخ.
7. Month: شهر التاريخ.
8. Year: سنة التاريخ.
9. PeriodOfDay: فترة نصف ساعة من اليوم.
10. ForcastWindProduction: توقعات إنتاج الرياح.
11. SystemLoadEA توقع الحمل الوطني المتوقع.
12. SMPEA: السعر المتوقع.
13. ORKTemperature: قياس درجة الحرارة الفعلية.
14. ORKWindspeed: قياس سرعة الرياح الفعلية.
15. CO2Intensity: الكثافة الفعلية لثاني أكسيد الكربون للكهرباء المنتجة.
16. ActualWindProduction: إنتاج طاقة الرياح الفعلي.
17. SystemLoadEP2: الحمل الفعلي للنظام الوطني.
18. SMPEP2: السعر الفعلي للكهرباء المستهلكة (علامات أو قيم يمكن توقعها).

لذا فإن مهمتك هنا هي استخدام هذه البيانات لتدريب نموذج التعلم الآلي للتنبؤ بسعر الكهرباء التي تستهلكها الآلات.في القسم أدناه، سأطلعك على مهمة التنبؤ بأسعار الكهرباء باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بأسعار الكهرباء باستخدام لغة بايثون

سأبدأ مهمة التنبؤ بأسعار الكهرباء عن طريق استيراد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لهذه المهمة:

```
import pandas as pd
import numpy as np
data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-
data/master/electricity.csv")
print(data.head())
```

```
           DateTime Holiday  ...  SystemLoadEP2  SMPEP2
0  01/11/2011 00:00    None  ...        3159.60   54.32
1  01/11/2011 00:30    None  ...        2973.01   54.23
2  01/11/2011 01:00    None  ...        2834.00   54.23
3  01/11/2011 01:30    None  ...        2725.99   53.47
4  01/11/2011 02:00    None  ...        2655.64   39.87

[5 rows x 18 columns]
```

دعونا نلقي نظرة على جميع أعمدة مجموعة البيانات هذه:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38014 entries, 0 to 38013
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   DateTime                38014 non-null  object
 1   Holiday                 38014 non-null  object
 2   HolidayFlag             38014 non-null  int64
 3   DayOfWeek               38014 non-null  int64
 4   WeekOfYear              38014 non-null  int64
 5   Day                     38014 non-null  int64
 6   Month                   38014 non-null  int64
 7   Year                    38014 non-null  int64
 8   PeriodOfDay             38014 non-null  int64
 9   ForecastWindProduction  38014 non-null  object
 10  SystemLoadEA            38014 non-null  object
 11  SMPEA                   38014 non-null  object
 12  ORKTemperature          38014 non-null  object
 13  ORKWindspeed            38014 non-null  object
 14  CO2Intensity            38014 non-null  object
 15  ActualWindProduction    38014 non-null  object
 16  SystemLoadEP2           38014 non-null  object
 17  SMPEP2                  38014 non-null  object
dtypes: int64(7), object(11)
memory usage: 5.2+ MB
```

أستطيع أن أرى أن العديد من الميزات ذات القيم العددية هي قيم سلسلةفي مجموعة البيانات وليست أعدادًا صحيحة أو قيمًا عائمة. لذا قبل المضي قدمًا، يتعين علينا تحويل قيم السلسلة النصية هذه إلى قيم عائمة float values:

```
data["ForecastWindProduction"] =
pd.to_numeric(data["ForecastWindProduction"], errors= 'coerce')
data["SystemLoadEA"] = pd.to_numeric(data["SystemLoadEA"], errors=
'coerce')
data["SMPEA"] = pd.to_numeric(data["SMPEA"], errors= 'coerce')
data["ORKTemperature"] = pd.to_numeric(data["ORKTemperature"], errors=
'coerce')
data["ORKWindspeed"] = pd.to_numeric(data["ORKWindspeed"], errors=
'coerce')
data["CO2Intensity"] = pd.to_numeric(data["CO2Intensity"], errors=
'coerce')
data["ActualWindProduction"] =
pd.to_numeric(data["ActualWindProduction"], errors= 'coerce')
data["SystemLoadEP2"] = pd.to_numeric(data["SystemLoadEP2"], errors=
'coerce')
data["SMPEP2"] = pd.to_numeric(data["SMPEP2"], errors= 'coerce')
```

دعنا الآن نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
DateTime                  0
Holiday                   0
HolidayFlag               0
DayOfWeek                 0
WeekOfYear                0
Day                       0
Month                     0
Year                      0
PeriodOfDay               0
ForecastWindProduction    5
SystemLoadEA              2
SMPEA                     2
ORKTemperature          295
ORKWindspeed            299
CO2Intensity              7
ActualWindProduction      5
SystemLoadEP2             2
SMPEP2                    2
dtype: int64
```

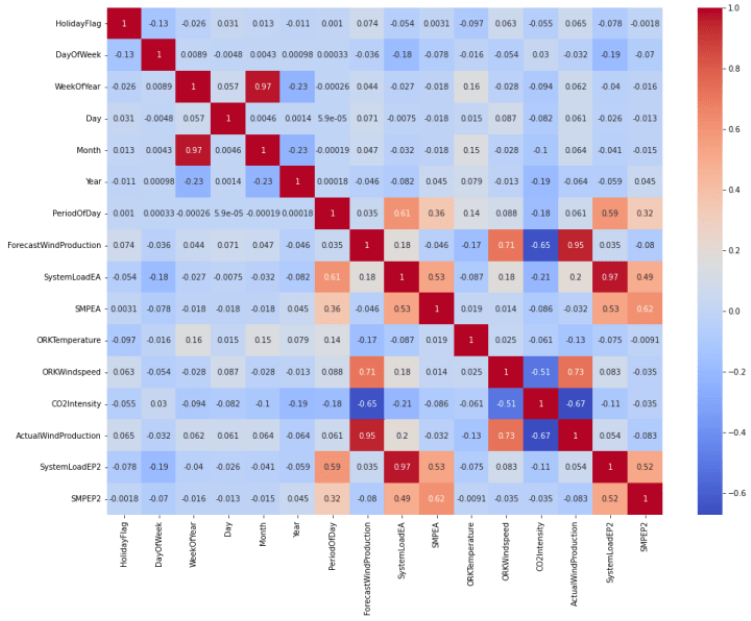
لذلك هناك بعض الأعمدة التي تحتوي على قيم خالية، سأقوم بإسقاط كل هذه الصفوف التي تحتوي على قيم خالية من مجموعة البيانات:

```
data = data.dropna()
```

دعنا الآن نلقي نظرة على الارتباط بين جميع الأعمدةفي مجموعة البيانات:

```
import seaborn as sns
import matplotlib.pyplot as plt
correlations = data.corr(method='pearson')
plt.figure(figsize=(16, 12))
```

```
sns.heatmap(correlations, cmap="coolwarm", annot=True)
plt.show()
```



نموذج التنبؤ بأسعار الكهرباء

دعنا الآن ننتقل إلى مهمة تدريب نموذج توقع أسعار الكهرباء. سأضيف هنا أولاً جميع الميزات المهمة إلى x والعمود المستهدف إلى y، وبعد ذلك سأقسم البيانات إلى مجموعات تدريب واختبار:

```
x = data[["Day", "Month", "ForecastWindProduction", "SystemLoadEA",
          "SMPEA", "ORKTemperature", "ORKWindspeed", "CO2Intensity",
          "ActualWindProduction", "SystemLoadEP2"]]
y = data["SMPEP2"]
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                test size=0.2,
                                                random_state=42)
```

نظرًا لأن هذه هي مشكلة الانحدار، سأختار هنا خوارزمية الغابة العشوائية (Random Forest) لتدريب نموذج التنبؤ بسعر الكهرباء:

```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
model.fit(xtrain, ytrain)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=100, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

دعنا الآن ندخل جميع قيم الميزات الضرورية التي استخدمناها لتدريب النموذج وإلقاء نظرة على سعر الكهرباء الذي تنبأ به النموذج:

```
#features = [["Day", "Month", "ForecastWindProduction",
"SystemLoadEA", "SMPEA", "ORKTemperature", "ORKWindspeed",
"CO2Intensity", "ActualWindProduction", "SystemLoadEP2"]]
features = np.array ,14.8 ,9.0 ,49.56 ,4241.05 ,54.10 ,12 ,10]])
([[4426.84 ,54.0 ,491.32
model.predict(features)
```

```
array([65.1696])
```

هذه هي الطريقة التي يمكنك بها تدريب نموذج التعلم الآلي للتنبؤ بأسعار الكهرباء.

الملخص

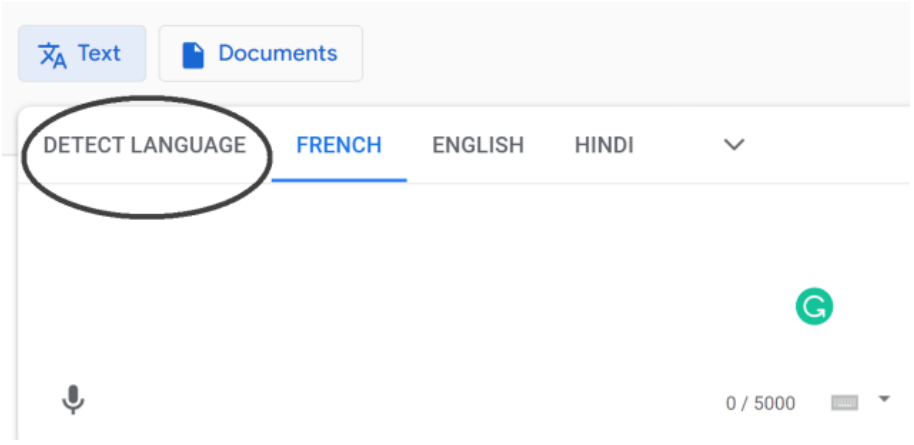
يساعد توقع سعر الكهرباء الكثير من الشركات على فهم مقدار نفقات الكهرباء التي يتعين عليهم دفعها كل عام. آمل أن تكون قد أحببت هذه المقالة حول مهمة التنبؤ بأسعار الكهرباء مع التعلم الآلي باستخدام بايثون.

23) اكتشاف اللغة باستخدام التعلم الآلي Language Detection with Machine Learning

يعد اكتشاف اللغة (Language detection) مهمة معالجة لغة طبيعية (natural language processing) حيث نحتاج إلى تحديد لغة النص أو المستند. كان استخدام التعلم الآلي لتحديد اللغة مهمة صعبة منذ بضع سنوات لأنه لم يكن هناك الكثير من البيانات حول اللغات، ولكن مع توفر البيانات بسهولة، تتوفر بالفعل العديد من نماذج التعلم الآلي القوية لتحديد اللغة. لذا، إذا كنت تريد معرفة كيفية تدريب نموذج التعلم الآلي على اكتشاف اللغة، فهذه المقالة مناسبة لك. في هذه المقالة، سوف أطلعك على مهمة اكتشاف اللغة باستخدام التعلم الآلي باستخدام بايثون.

اكتشاف اللغة

كإنسان، يمكنك بسهولة اكتشاف اللغات التي تعرفها. على سبيل المثال، يمكنني التعرف بسهولة على الهندية والإنجليزية، لكن كوني هنديًا، لا يمكنني أيضًا تحديد جميع اللغات الهندية. هذا هو المكان الذي يمكن فيه استخدام مهمة تحديد اللغة. يعد Google Translate أحد أشهر مترجمي اللغاتفي العالم ويستخدمه الكثير من الأشخاص حول العالم. يتضمن أيضًا نموذجًا للتعلم الآلي لاكتشاف اللغات التي يمكنك استخدامها إذا كنت لا تعرف اللغة التي تريد ترجمتها.



البيانات (Data) هي أهم جزءفي تدريب نموذج الكشف عن اللغة. كلما زادت البيانات المتوفرة لديك حول كل لغة، زادت دقة أداء نموذجكفي الوقت الفعلي. تم جمع مجموعة البيانات التي أستخدمها من Kaggle، والتي تحتوي على بيانات حول 22 لغة شائعة وتحتوي على 1000 جملةفي كل لغة، لذلك ستكون مجموعة بيانات مناسبة لتدريب نموذج اكتشاف اللغة باستخدام التعلم الآلي. لذلكفي القسم أدناه، سأطلعك على كيفية تدريب نموذج اكتشاف اللغة باستخدام التعلم الآلي باستخدام بايثون.

اكتشاف اللغة باستخدام بايثون

لنبدأ مهمة اكتشاف اللغة باستخدام التعلم الآلي عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/dataset.csv")
print(data.head())
```

```
                                                Text  language
0  klement gottwaldi surnukeha palsameeriti ning ...  Estonian
1  sebes joseph pereira thomas  på eng the jesuit...   Swedish
2  ถนนเจริญกรุง อักษรโรมัน thanon charoen krung ไ...      Thai
3  விசாகப்பட்டினம் தமிழ்ச்சங்கத்தை இந்துப் பத்திர...     Tamil
4  de spons behoort tot het geslacht haliclona en...     Dutch
```

دعونا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
Text        0
language    0
dtype: int64
```

دعنا الآن نلقي نظرة على جميع اللغات الموجودةفي مجموعة البيانات هذه:

```
data["language"].value_counts()
```

```
Persian       1000
Korean        1000
Indonesian    1000
Dutch         1000
Estonian      1000
Hindi         1000
Urdu          1000
French        1000
Turkish       1000
Latin         1000
Arabic        1000
Russian       1000
Thai          1000
Spanish       1000
English       1000
Japanese      1000
Pushto        1000
Chinese       1000
Tamil         1000
Swedish       1000
Romanian      1000
Portugese     1000
Name: language, dtype: int64
```

تحتوي مجموعة البيانات هذه على 22 لغة مع 1000 جملة من كل لغة. هذه مجموعة بيانات متوازنة للغاية ولا تحتوي على قيم مفقودة، لذلك يمكننا القول إن مجموعة البيانات هذه جاهزة تمامًا لاستخدامهافي تدريب نموذج التعلم الآلي.

نموذج اكتشاف اللغة

دعنا الآن نقسم البيانات إلى مجموعات تدريب واختبار:

```
x = np.array(data["Text"])
y = np.array(data["language"])

cv = CountVectorizer()
X = cv.fit_transform(x)
X_train, X_test, y_train, y_test = train_test_split(X, y,
                    test_size=0.33,
                    random_state=42)
```

نظرًا لأن هذه مشكلة تتعلق بتصنيف متعدد الفئات (multiclass classification)، لذلك سأستخدم خوارزمية Multinomial Naïve Bayes لتدريب نموذج اكتشاف اللغة لأن هذه الخوارزمية تعمل دائمًا بشكل جيد جدًا على المشكلات القائمة على تصنيف متعدد الفئات:

```
model = MultinomialNB()
model.fit(X_train,y_train)
model.score(X_test,y_test)
```

```
0.953168044077135
```

دعنا الآن نستخدم هذا النموذج لاكتشاف لغة النص من خلال أخذ إدخال المستخدم:

```
user = input("Enter a Text: ")
data = cv.transform([user]).toarray()
output = model.predict(data)
print(output)
```

```
Enter a Text: देखकर अच्छा लगता है
['Hindi']
```

كما ترون أن النموذج يعمل بشكل جيد. شيء واحد يجب ملاحظته هنا هو أن هذا النموذج يمكنه فقط اكتشاف اللغات المذكورةفي مجموعة البيانات.

الملخص

كان استخدام التعلم الآلي لتحديد اللغة مهمة صعبة منذ بضع سنوات لأنه لم يكن هناك الكثير من البيانات حول اللغات، ولكن مع توفر البيانات بسهولة، تتوفر بالفعل العديد من نماذج التعلم

الآلي القوية لتحديد اللغة. آمل أن تكون قد أحببت هذه المقالة حول اكتشاف اللغات باستخدام التعلم الآلي باستخدام بايثون.

24) التنبؤ بسعر سهم Tata Motors مع التعلم الآلي Tata Motors Stock Price Prediction with Machine Learning

في الآونة الأخيرة، شهدنا زيادة بأكثر من 10في المائةفي سعر سهم شركة Tata Motors. وقد أدى ذلك إلى مزيد من الاهتمام بأسهم مجموعة تاتا من جميع أنحاء الهند. ولكن مرة أخرى نشهد اليوم انخفاضًافي أسعار أسهم شركة Tata Motors، وهو ما يمكن أن يكون إشارة سلبية للمستثمرين. لذا، إذا كنت تريد معرفة كيفية تحليل سعر سهم شركة Tata Motors والتنبؤ به، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة التنبؤ بأسعار أسهم شركة Tata Motors مع التعلم الآلي باستخدام بايثون.

التنبؤ بسعر سهم Tata Motors

لمهمة التنبؤ بأسعار أسهم شركة Tata Motors، تحتاج إلى تنزيل مجموعة بيانات أسعار أسهم شركة Tata Motors. لذا، لتنزيل أحدث مجموعة بيانات لأسعار الأسهم، ما عليك سوى اتباع الخطوات المذكورة أدناه:

1. قم بزيارة Yahoo Finance.
2. ابحث عن Tata Motors أو TTM (رمز سهم شركة Tata Motors)
3. ثم انقر فوق البيانات التاريخية.
4. وفي النهاية انقر فوق "download ".

بعد هذه الخطوات، سترى ملف CSVفي مجلد التنزيلات الخاص بك. الآنفي القسم أدناه، سوف آخذك خلال مهمة توقع سعر سهم Tata Motors مع التعلم الآلي باستخدام بايثون.

التنبؤ بأسعار أسهم شركة Tata Motors باستخدام لغة بايثون

لنبدأ مهمة التنبؤ بأسعار أسهم شركة Tata Motors عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import numpy as np
import pandas as pd
import plotly.graph_objects as go
data = pd.read_csv("TTM.csv")
print(data.head())
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2020-10-22	9.08	9.14	9.03	9.12	9.12	1049600
1	2020-10-23	9.30	9.34	9.16	9.32	9.32	1158700
2	2020-10-26	9.03	9.08	8.89	9.06	9.06	2141900
3	2020-10-27	9.15	9.58	9.15	9.48	9.48	1552900
4	2020-10-28	9.02	9.04	8.80	8.85	8.85	2483900

دعنا الآن نرسم تصويرًا تفاعليًا لأسعار الأسهم للحصول على صورة واضحة لزيادة وانخفاض أسعار أسهم شركة Tata Motors:

```
figure = go.Figure(data=[go.Candlestick(x=data["Date"],
                                        open=data["Open"],
high=data["High"],
                                        low=data["Low"],
close=data["Close"])])
figure.update_layout(title = "Tata Motors Stock Price
Analysis", xaxis_rangeslider_visible=False)
figure.show()
```

دعنا الآن نلقي نظرة على الارتباط (correlation) بين ميزات مجموعة البيانات هذه:

```
print(data.corr())
```

	Open	High	Low	Close	Adj Close	Volume
Open	1.000000	0.998775	0.999278	0.998043	0.998043	0.106946
High	0.998775	1.000000	0.998695	0.999452	0.999452	0.132946
Low	0.999278	0.998695	1.000000	0.998787	0.998787	0.100552
Close	0.998043	0.999452	0.998787	1.000000	1.000000	0.127651
Adj Close	0.998043	0.999452	0.998787	1.000000	1.000000	0.127651
Volume	0.106946	0.132946	0.100552	0.127651	0.127651	1.000000

دعنا الآن ننتقل إلى مهمة التنبؤ بأسعار أسهم شركة Tata Motors. سأستخدم هنا مكتبة autots في بايثون لإعداد أسعار أسهم شركة Tata Motors للأيام الخمسة القادمة. إذا لم تستخدم مكتبة بايثون هذه من قبل، فيمكنك تثبيتها بسهولة باستخدام الأمر pip:

```
pip install autots
```

الآن فيما يلي كيف يمكنك توقع أسعار أسهم شركة Tata Motors:

```
from autots import AutoTS
model = AutoTS(forecast_length=5, frequency='infer',
ensemble='simple')
model = model.fit(data, date_col='Date', value_col='Close',
id_col=None)
prediction = model.predict()
forecast = prediction.forecast
print(forecast)
```

```
              Close
2021-10-22  34.060566
2021-10-25  34.525269
2021-10-26  34.955687
2021-10-27  35.478639
2021-10-28  35.850695
```

هذه هي الطريقة التي يمكنك بها استخدام التعلم الآلي لمهمة التنبؤ بسعر سهم شركة Tata Motors.

الملخص

هذه هي الطريقة التي يمكنك من خلالها توقع أسعار أسهم شركة Tata Motors من خلال التعلم الآلي. تحظى شركة Tata Motors باهتمام كبيرفي سوق الأسهم، لذلك سيكون هذا هو أفضل وقت لتحليل أسعار أسهم شركة Tata Motors. آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بأسعار أسهم شركة Tata Motors مع التعلم الآلي باستخدام بايثون.

25) التنبؤ بأقساط التأمين الصحي باستخدام التعلم الآلي Health Insurance Premium Prediction with Machine Learning

التأمين الصحي (Health Insurance) هو نوع من التأمين يغطي النفقات الطبية. يحصل الشخص الذي حصل على بوليصة تأمين صحي على تغطية تأمين صحي بدفع مبلغ قسط معين. هناك الكثير من العوامل التي تحدد قسط التأمين الصحي. لذلك إذا كنت تريد معرفة كيف يمكننا استخدام التعلم الآلي للتنبؤ بقسط التأمين الصحي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بأقساط التأمين الصحي مع التعلم الآلي باستخدام بايثون.

التنبؤ بأقساط التأمين الصحي

يعتمد مبلغ قسط بوليصة التأمين الصحي من شخص لآخر، حيث تؤثر العديد من العوامل على مبلغ قسط بوليصة التأمين الصحي. لنفترض أن العمر، الشاب أقل عرضة للإصابة بمشاكل صحية كبيرة مقارنة بالشخص الأكبر سنًا. وبالتالي، فإن علاج كبار السن سيكون مكلفًا مقارنة بالشباب. هذا هو السببفي أن الشخص الأكبر سنًا مطالب بدفع أقساط عالية مقارنة بالشاب.

تمامًا مثل العمر، هناك العديد من العوامل الأخرى التي تؤثر على قسط التأمين الصحي. آمل أن تكون قد فهمت الآن ماهية التأمين الصحي وكيف يتم تحديد قسط التأمين الصحي.في القسم أدناه، سوف آخذك خلال مهمة التنبؤ بأقساط التأمين الصحي باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بأقساط التأمين الصحي باستخدام بايثون

يتم جمع مجموعة البيانات التي أستخدمها لمهمة التنبؤ بأقساط التأمين الصحي من Kaggle. يحتوي على بيانات حول:

1. عمر الشخص (the age of the person).
2. جنس الشخص (gender of the person).
3. مؤشر كتلة الجسم للشخص (Body Mass Index of the person).
4. كم عدد الأطفال الذين ينجبهم الشخص (how many children the person is having).
5. سواء كان الشخص يدخن أم لا (whether the person smokes or not).
6. المنطقة التي يعيش فيها الشخص (the region where the person lives).

7. ورسوم قسط التأمين (the charges of the insurance premium).

لذلك دعونا نستورد مجموعة البيانات ومكتبات بايثون الضرورية التي نحتاجها لهذه المهمة:

```
import numpy as np
import pandas as pd
data = pd.read_csv("Health_insurance.csv")
data.head()
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

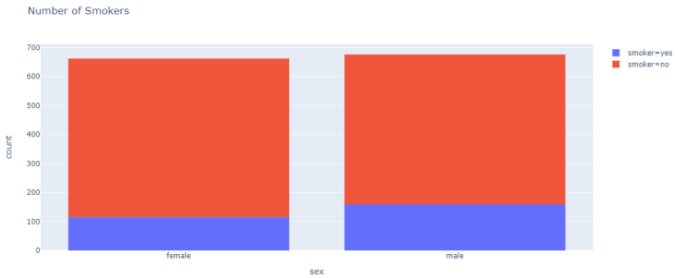
قبل المضي قدمًا، دعنا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

لذلك فإن مجموعة البيانات جاهزة للاستخدام. بعد الحصول على الانطباعات الأولى عن هذه البيانات، لاحظت عمود المدخن (smoker) الذي يشير إلى ما إذا كان الشخص يدخن أم لا. هذه ميزة مهمة لمجموعة البيانات هذه لأن الشخص الذي يدخن يكون أكثر عرضة للإصابة بمشاكل صحية كبيرة مقارنة بالشخص الذي لا يدخن. فلنلق نظرة على توزيع الأشخاص الذين يدخنون والذين لا يدخنون:

```
import plotly.express as px
data = data
figure = px.histogram(data, x = "sex", color = "smoker",
title= "Number of Smokers")
figure.show()
```



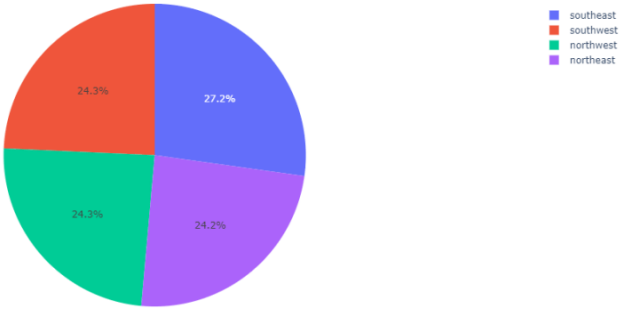
حسب الرسم أعلاه، 547 أنثى، 517 ذكر لا يدخنون، و115 أنثى، 159 ذكر يدخنون. من المهم استخدام هذه الميزة أثناء تدريب نموذج التعلم الآلي، لذلك سأستبدل قيم عمودي الجنس (Sex) والمدخن (smoker) بـ 0 و 1 لأن كلا العمودين يحتويان على قيم سلسلة نصية:

```
data["sex"] = data["sex"].map({"female": 0, "male": 1})
data["smoker"] = data["smoker"].map({"no": 0, "yes": 1})
print(data.head())
```

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	southwest	16884.92400
1	18	1	33.770	1	0	southeast	1725.55230
2	28	1	33.000	3	0	southeast	4449.46200
3	33	1	22.705	0	0	northwest	21984.47061
4	32	1	28.880	0	0	northwest	3866.85520

دعنا الآن نلقي نظرة على توزيع المناطق (distribution of the regions) التي يعيش فيها الأشخاص وفقًا لمجموعة البيانات:

```
import plotly.express as px
pie = data["region"].value_counts()
regions = pie.index
population = pie.values
fig = px.pie(data, values=population, names=regions)
fig.show()
```



دعنا الآن نلقي نظرة على الارتباط (correlation) بين ميزات مجموعة البيانات هذه:

	age	sex	bmi	children	smoker	charges
age	1.000000	-0.020856	0.109272	0.042469	-0.025019	0.299008
sex	-0.020856	1.000000	0.046371	0.017163	0.076185	0.057292
bmi	0.109272	0.046371	1.000000	0.012759	0.003750	0.198341
children	0.042469	0.017163	0.012759	1.000000	0.007673	0.067998
smoker	-0.025019	0.076185	0.003750	0.007673	1.000000	0.787251
charges	0.299008	0.057292	0.198341	0.067998	0.787251	1.000000

نموذج التنبؤ بأقساط التأمين الصحي

دعنا الآن ننتقل إلى تدريب نموذج التعلم الآلي لمهمة التنبؤ بأقساط التأمين الصحي. أولاً، سأقسم البيانات إلى مجموعات تدريب واختبار:

```
x = np.array(data[["age", "sex", "bmi", "smoker"]])
y = np.array(data["charges"])

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.2, random_state=42)
```

بعد استخدام خوارزميات مختلفة للتعلم الآلي، وجدت خوارزمية الغابة العشوائية (random forest) كأفضل خوارزمية أداء لهذه المهمة. لذلك سأقوم هنا بتدريب النموذج باستخدام خوارزمية الغابة العشوائية للانحدار:

```
from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor()
forest.fit(xtrain, ytrain)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=100, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

دعنا الآن نلقي نظرة على القيم المتوقعة (predicted values) للنموذج:

```
ypred = forest.predict(xtest)
data = pd.DataFrame(data={"Predicted Premium Amount": ypred})
print(data.head())
```

```
   Predicted Premium Amount
0              11331.111753
1               5366.132261
2              28257.205036
3               9793.356425
4              34720.204296
```

هذه هي الطريقة التي يمكنك بها تدريب نموذج التعلم الآلي لمهمة التنبؤ بأقساط التأمين الصحي باستخدام بايثون.

الملخص

يعتمد مبلغ قسط بوليصة التأمين الصحي على شخص لآخر حيث أن العديد من العوامل تؤثر على مبلغ قسط بوليصة التأمين الصحي. آمل أن تكون قد أحببت هذه المقالة حول تنبؤات أقساط التأمين الصحي باستخدام التعلم الآلي باستخدام بايثون.

26) التنبؤ بعدد الطلبات باستخدام التعلم الآلي Number of Orders Prediction with Machine Learning

يعد التنبؤ بعدد الطلبات (Predicting the number of orders) لمنتج ما إحدى الاستراتيجيات التي يمكن أن تتبعها الشركةفي تحديد مقدار الاستثمارفي تسويق منتجها. لذا، فإن التنبؤ بعدد الطلبات هو حالة مهمة لاستخدام علم البيانات للشركات القائمة على المنتجات. إذا كنت تريد معرفة كيفية استخدام التعلم الآلي للتنبؤ بعدد الطلبات، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة التنبؤ بعدد الطلبات باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بعدد الطلبات

إذا كنت ترغبفي التنبؤ بعدد الطلبات التي قد تتلقاها الشركة لمنتج معين، فأنت بحاجة إلى الحصول على بيانات تاريخية حول عدد الطلبات التي تتلقاها الشركة. لذلكفي هذه المهمة، سأستخدم بيانات مبيعات المكملات الغذائية (supplements) التي تم جمعها من Kaggle. البيانات التي سأستخدمها لمهمة التنبؤ بعدد الطلبات تحتوي على بيانات حول:

1. معرف المنتج (Product ID).
2. معرف المتجر (Store ID).
3. نوع المتجر الذي تم بيع المكمل فيه (The type of store where the supplement was sold).
4. نوع الموقع الذي تم استلام الطلب منه (The type of location the order was received from).
5. تاريخ المبيعات (Sales Date).
6. رمز المنطقة (Region code).
7. سواء كانت عطلة عامة أم لا وقت الطلب (Whether it is a public holiday or not at the time of order).
8. سواء كان المنتج بسعر مخفض أم لا (Whether the product was on discount or not).

9. عدد الطلبات المقدمة (Number of orders placed).

10. المبيعات (Sales).

آمل أن تكون قد حصلت الآن على نظرة عامة حول المشكلة ومجموعة البيانات التي سأستخدمها لحل المشكلة. الآنفي القسم أدناه، سوف آخذك خلال مهمة التنبؤ بعدد الطلبات باستخدام التعلم الآلي باستخدام لغة برمجة بايثون.

التنبؤ بعدد الطلبات باستخدام بايثون

لنبدأ مهمة التنبؤ بعدد الطلبات عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/supplement.csv")
data.head()
```

```
         ID  Store_id Store_Type  ... Discount #Order     Sales
0  T1000001         1         S1  ...      Yes      9   7011.84
1  T1000002       253         S4  ...      Yes     60  51789.12
2  T1000003       252         S3  ...      Yes     42  36868.20
3  T1000004       251         S2  ...      Yes     23  19715.16
4  T1000005       250         S2  ...      Yes     62  45614.52

[5 rows x 10 columns]
```

دعنا الآن نلقي نظرة على بعض الأفكار الضرورية من مجموعة البيانات هذه لمعرفة نوع مجموعة البيانات التي نعمل معها:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188340 entries, 0 to 188339
Data columns (total 10 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   ID             188340 non-null  object
 1   Store_id       188340 non-null  int64
 2   Store_Type     188340 non-null  object
 3   Location_Type  188340 non-null  object
 4   Region_Code    188340 non-null  object
 5   Date           188340 non-null  object
 6   Holiday        188340 non-null  int64
 7   Discount       188340 non-null  object
 8   #Order         188340 non-null  int64
 9   Sales          188340 non-null  float64
dtypes: float64(1), int64(3), object(6)
memory usage: 14.4+ MB
```

```
data.isnull().sum()
```

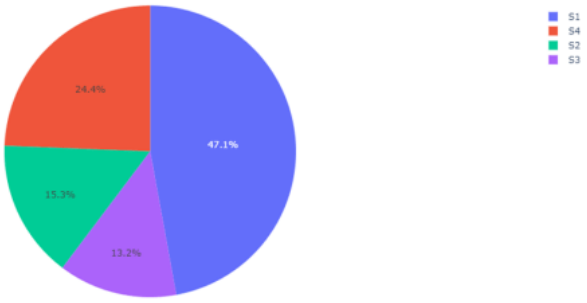
```
ID              0
Store_id        0
Store_Type      0
Location_Type   0
Region_Code     0
Date            0
Holiday         0
Discount        0
#Order          0
Sales           0
dtype: int64
```

```
data.describe()
```

	Store_id	Holiday	#Order	Sales
count	188340.000000	188340.000000	188340.000000	188340.000000
mean	183.000000	0.131783	68.205692	42784.327982
std	105.366308	0.338256	30.467415	18456.708302
min	1.000000	0.000000	0.000000	0.000000
25%	92.000000	0.000000	48.000000	30426.000000
50%	183.000000	0.000000	63.000000	39678.000000
75%	274.000000	0.000000	82.000000	51909.000000
max	365.000000	1.000000	371.000000	247215.000000

دعنا الآن نستكشف بعض الميزات المهمة من مجموعة البيانات هذه للتعرف على العوامل التي تؤثر على عدد طلبات المكملات:

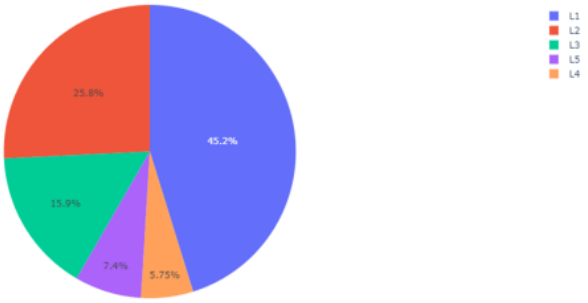
```
import plotly.express as px
pie = data["Store_Type"].value_counts()
store = pie.index
orders = pie.values

fig = px.pie(data, values=orders, names=store)
fig.show()
```



يوضح الشكل أعلاه توزيع عدد الطلبات المستلمة حسب نوع المتجر. دعنا الآن نلقي نظرة على توزيع عدد الطلبات، حسب الموقع:

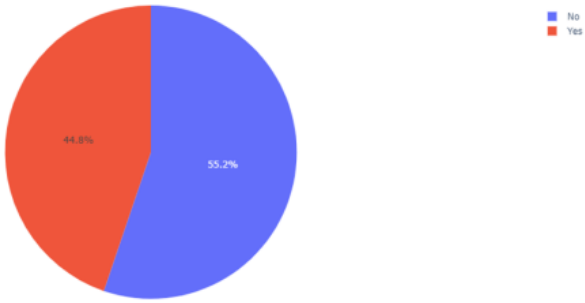
```
pie2 = data["Location_Type"].value_counts()
location = pie2.index
orders = pie2.values
```

```
fig = px.pie(data, values=orders, names=location)
fig.show()
```



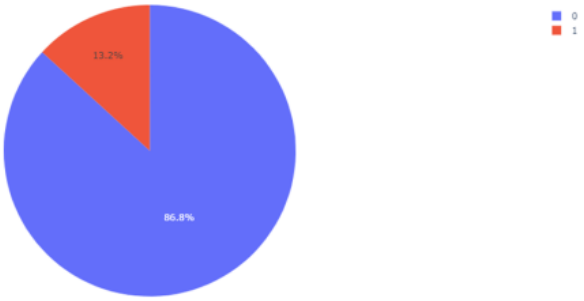
يوضح الشكل أعلاه توزيع عدد الطلبات المستلمة وفقًا للموقع. دعنا الآن نلقي نظرة على توزيع عدد الطلبات، وفقًا للخصم:

```
pie3 = data["Discount"].value_counts()
discount = pie3.index
orders = pie3.values

fig = px.pie(data, values=orders, names=discount)
fig.show()
```



وفقًا للرقم أعلاه، لا يزال معظم الناس يشترون المكملات الغذائية إذا لم يكن هناك خصم عليها. دعنا الآن نلقي نظرة على كيفية تأثير العطلات على عدد الطلبات:

```
pie4 = data["Holiday"].value_counts()
holiday = pie4.index
orders = pie4.values

fig = px.pie(data, values=orders, names=holiday)
fig.show()
```



وفقًا للرقم أعلاه، يشتري معظم الناس المكملات الغذائيةفي أيام العمل.

نموذج التنبؤ بعدد الطلبات

الآن دعنا نجهز البيانات حتى نتمكن من تدريب نموذج التعلم الآلي لمهمة التنبؤ بعدد الطلبات.

هنا، سوف أقوم بتغيير بعض قيم السلسلة النصية إلى قيم رقمية:

```
data["Discount"] = data["Discount"].map({"No": 0, "Yes": 1})
data["Store_Type"] = data["Store_Type"].map({"S1": 1, "S2": 2,
"S3": 3, "S4": 4})
data["Location_Type"] = data["Location_Type"].map({"L1": 1,
"L2": 2, "L3": 3, "L4": 4, "L5": 5})
data.dropna()

x = np.array(data[["Store_Type", "Location_Type", "Holiday",
"Discount"]])
y = np.array(data["#Order"])
```

الآن دعنا نقسم البيانات إلى مجموعة تدريب 80٪ ومجموعة اختبار 20٪:

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x,
                                                y,
test_size=0.2, random_state=42)
```

سأستخدم الآن خوارزمية الانحدار المعزز للتدرج الخفيف (light gradient boosting regression) لتدريب النموذج:

```
# Use pip install lightgbm to install it on your system
import lightgbm as ltb
model = ltb.LGBMRegressor()
model.fit(xtrain, ytrain)
```

```
LGBMRegressor(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
              importance_type='split', learning_rate=0.1, max_depth=-1,
              min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
              n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
              random_state=None, reg_alpha=0.0, reg_lambda=0.0, silent=True,
              subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
```

الآن دعنا نلقي نظرة على القيم المتوقعة (predicted values):

```
ypred = model.predict(xtest)
data = pd.DataFrame(data={"Predicted Orders": ypred.flatten()})
print(data.head())
```

```
   Predicted Orders
0         47.351897
1         97.068717
2         66.577788
3         85.143083
4         54.451098
```

هذه هي الطريقة التي يمكنك بها تدريب نموذج التعلم الآلي على مهمة التنبؤ بعدد الطلبات باستخدام لغة برمجة بايثون.

الملخص

يعد التنبؤ بعدد طلبات المنتج أحد الاستراتيجيات التي يمكن لشركة قائمة على المنتج اتباعها لتحديد المبلغ الذي يجب عليهم استثمارهفي تسويق منتجهم. آمل أن تكون قد أحببت هذه المقالة حول مهمة التنبؤ بعدد الطلبات باستخدام التعلم الآلي باستخدام بايثون.

27) التنبؤ بسعر سهم Apple مع التعلم الآلي Apple Stock Price Prediction with Machine Learning

أعلنت شركة Apple للتو عن موعد حدث سبتمبر حيث توشك على إطلاق iPhone 13 الجديد. وهو حاليًا مركز الاهتمامفي سوق الأسهم. يعد تحليل سوق الأوراق المالية أحد التطبيقات الشائعة للتعلم الآلي لأنه يمكننا التنبؤ بأسعار الأسهم باستخدام التعلم الآلي. لذلك إذا كنت تريد معرفة كيفية التنبؤ بأسعار أسهم Apple باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة توقع أسعار أسهم Apple مع التعلم الآلي باستخدام بايثون.

التنبؤ بسعر سهم Apple

يعد حدث Appleفي سبتمبر أحد الأحداث المفضلة لجميع مستخدمي Apple، حيث يتم إطلاق أجهزة iPhone بشكل أساسي خلال حدث سبتمبر. لذلك أعلنت شركة آبل أنها ستستعد لإطلاق iPhone 13 الجديدفي 14 سبتمبر. لذلك يمكن للعديد من المستثمرينفي سوق الأسهم أن يجدوا هذا كفرصة لشراء أسهم Apple، لأنهفي كل مرة تأتي فيها شركة بمنتج مبتكر، فإنها تقود لزيادة سعر سهمها. ومع أخذ ذلكفي الاعتبار، يمكننا القول إن هذا هو أفضل وقت لتحليل أسعار أسهم Apple.

بالنسبة لمهمة التنبؤ بسعر سهم Apple، فأنت بحاجة إلى تنزيل مجموعة بيانات أسعار أسهم Apple. لتنزيل مجموعة بيانات لهذه المهمة، اتبع الخطوات المذكورة أدناه:

1. قم بزيارة Yahoo Finance.
2. ابحث عن Apple أو AAPL (رمز سهم Apple).
3. ثم انقر فوق البيانات التاريخية.
4. وفي النهاية انقر فوق "download".

بعد هذه الخطوات، سترى ملف CSVفي مجلد التنزيل الخاص بك. الآن،في القسم أدناه، سوف أطلعك على مهمة توقع سعر سهم Apple باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بسعر سهم Apple باستخدام بايثون

لنبدأ مهمة التنبؤ بأسعار أسهم Apple عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import numpy as np
```

```
import pandas as pd
import plotly.graph_objects as go
data = pd.read_csv("AAPL.csv")
print(data.head())
```

```
         Date        Open        High  ...       Close   Adj Close     Volume
0  2020-09-08  113.949997  118.989998  ...  112.820000  112.098999  231366600
1  2020-09-09  117.260002  119.139999  ...  117.320000  116.570236  176940500
2  2020-09-10  120.360001  120.500000  ...  113.489998  112.764717  182274400
3  2020-09-11  114.570000  115.230003  ...  112.000000  111.284241  180860300
4  2020-09-14  114.720001  115.930000  ...  115.360001  114.622765  140150100

[5 rows x 7 columns]
```

الآن دعنا نتخيل بيانات أسعار الأسهم هذه للحصول على صورة واضحة لزيادة وانخفاض أسعار أسهم Apple:

```
figure = go.Figure(data=[go.Candlestick(x=data["Date"],
                                        open=data["Open"],
high=data["High"],
                                        low=data["Low"],
close=data["Close"])])
figure.update_layout(title = "Apple Stock Price Analysis",
xaxis_rangeslider_visible=False)
figure.show()
```

دعنا الآن نلقي نظرة على الارتباط (correlation) بين الميزاتفي مجموعة البيانات هذه:

```
print(data.corr())
```

```
                Open      High       Low     Close  Adj Close    Volume
Open        1.000000  0.994551  0.993183  0.986214   0.986177 -0.466464
High        0.994551  1.000000  0.992951  0.993586   0.993307 -0.440943
Low         0.993183  0.992951  1.000000  0.993915   0.994187 -0.517453
Close       0.986214  0.993586  0.993915  1.000000   0.999899 -0.489536
Adj Close   0.986177  0.993307  0.994187  0.999899   1.000000 -0.493909
Volume     -0.466464 -0.440943 -0.517453 -0.489536  -0.493909  1.000000
```

دعنا الآن ننتقل إلى مهمة التنبؤ بأسعار أسهم Apple. سأستخدم هنا مكتبة autotsفي بايثون للتنبؤ بأسعار أسهم Apple للأيام الخمسة القادمة. إذا لم تستخدمه من قبل، فيمكنك تثبيته بسهولة باستخدام الأمر pip:

```
pip install autots
```

الآن فيما يلي كيف يمكنك توقع أسعار أسهم شركة Apple:

```
from autots import AutoTS
model = AutoTS(forecast_length=5, frequency='infer',
ensemble='simple')
model = model.fit(data, date_col='Date', value_col='Close',
id_col=None)
prediction = model.predict()
forecast = prediction.forecast
print(forecast)
```

```
                Close
2021-09-08  157.595000
2021-09-09  158.491248
2021-09-10  157.846256
2021-09-13  158.758755
2021-09-14  159.934376
```

هذه هي الطريقة التي يمكنك بها استخدام التعلم الآلي للتنبؤ بأسعار الأسهم.

الملخص

هذه هي الطريقة التي يمكنك بها التنبؤ بأسعار أسهم Apple باستخدام التعلم الآلي باستخدام لغة برمجة بايثون. يعد تحليل سوق الأوراق المالية أحد التطبيقات الشائعة للتعلم الآلي لأنه يمكننا التنبؤ بأسعار الأسهم باستخدام التعلم الآلي. آمل أن تكون قد أحببت هذه المقالة حول توقع أسعار أسهم Apple مع التعلم الآلي باستخدام بايثون.

28) تصنيف الأخبار مع تعلم الآلة News Classification with Machine Learning

يجب أن تكون قد شاهدت الأخبار مقسمة إلى فئات عندما تذهب إلى موقع إخباري. بعض الفئات الشائعة التي ستشاهدهافي أي موقع إخباري تقريبًا هي التكنولوجيا والترفيه والرياضة. إذا كنت تريد معرفة كيفية تصنيف فئات الأخبار باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تصنيف الأخبار باستخدام التعلم الآلي باستخدام بايثون.

تصنيف الأخبار

يصنف كل موقع إخباري المقال الإخباري قبل نشره بحيث يمكن للزائرينفي كل مرة يزورون موقع الويب الخاص بهم النقر بسهولة على نوع الأخبار التي تهمهم. على سبيل المثال، أحب قراءة آخر تحديثات التكنولوجيا، لذلكفي كل مرة أزور فيها موقعًا إخباريًا، أنقر على قسم التكنولوجيا. ولكن قد ترغب أو لا ترغبفي القراءة عن التكنولوجيا، فقد تكون مهتمًا بالسياسة أو الأعمال أو الترفيه أو ربما الرياضة.

حاليًا، يتم تصنيف المقالات الإخبارية يدويًا بواسطة مديري المحتوىفي مواقع الويب الإخبارية. ولكن لتوفير الوقت، يمكنهم أيضًا تنفيذ نموذج التعلم الآلي على مواقع الويب الخاصة بهم والذي يقرأ عنوان الأخبار أو محتوى الأخبار ويصنف فئة الأخبار.في القسم أدناه، سأطلعك على كيفية تدريب نموذج التعلم الآلي لمهمة تصنيف الأخبار باستخدام لغة برمجة بايثون.

تصنيف الأخبار باستخدام بايثون

بالنسبة لمهمة تصنيف الأخبار باستخدام التعلم الآلي، فقد جمعت مجموعة بيانات من Kaggle، والتي تحتوي على مقالات إخبارية بمافي ذلك عناوينها وفئاتها. الفئات المشمولةفي مجموعة البيانات هذه هي:

1. رياضة (Sports).
2. اعمال (Business).
3. سياسة (Politics).
4. تقنية (Tech).
5. الترفيه (Entertainment).

لذلك دعونا نستورد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لهذه المهمة:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/bbc-news-data.csv", sep='\t')
print(data.head())
```

```
   category  ...                                            content
0  business  ...  Quarterly profits at US media giant TimeWarne...
1  business  ...  The dollar has hit its highest level against ...
2  business  ...  The owners of embattled Russian oil giant Yuk...
3  business  ...  British Airways has blamed high fuel prices f...
4  business  ...  Shares in UK drinks and food firm Allied Dome...

[5 rows x 4 columns]
```

الآن، دعنا نلقي نظرة سريعة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
category    0
filename    0
title       0
content     0
dtype: int64
```

توجد التسميات (labels) التي نحتاج إلى تصنيفها من مجموعة البيانات هذهفي عمود الفئة (category column) بهذه البيانات، دعنا نلقي نظرة على توزيع جميع فئات الأخبار:

```
data["category"].value_counts()
```

```
sport          511
business       510
politics       417
tech           401
entertainment  386
Name: category, dtype: int64
```

نموذج تصنيف الأخبار

دعنا الآن نجهز البيانات لمهمة تدريب نموذج تصنيف الأخبار:

```
data = data[["title", "category"]]

x = np.array(data["title"])
y = np.array(data["category"])

cv = CountVectorizer()
X = cv.fit_transform(x)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)
```

سأستخدم الآن خوارزمية Multinomial Naive Bayes لتدريب نموذج تصنيف الأخبار:

```
model = MultinomialNB()
model.fit(X_train,y_train)
```

أخيرًا، دعنا نختبر كيفية عمل هذا النموذج على أحد العناوين الرئيسية (headlines)في أخبار اليوم:

```
user = input("Enter a Text: ")
data = cv.transform([user]).toarray()
output = model.predict(data)
print(output)
```

```
Enter a Text: Latest Apple iPhone SE 3 concept renders show a compact smartphone in the style of the
iPhone 4
['tech']
```

هذه هي الطريقة التي يمكنك بها تدريب نموذج تصنيف الأخبار باستخدام التعلم الآلي باستخدام بايثون.

الملخص

هذه هي الطريقة التي يمكننا بها استخدام التعلم الآلي لتصنيف فئات الأخبار. يصنف كل موقع إخباري المقال الإخباري قبل نشره بحيث يمكن للزائرين في كل مرة يزورون موقع الويب الخاص بهم النقر بسهولة على نوع الأخبار التي تهمهم. آمل أن تكون قد أحببت هذه المقالة حول تصنيف الأخبار مع التعلم الآلي باستخدام بايثون.

29) التنبؤ بأسعار السيارة مع التعلم الآلي Car Price Prediction with Machine Learning

يعتمد سعر السيارة (Car Price) على الكثير من العوامل مثل السمعة الحسنة للعلامة التجارية للسيارة وميزات السيارة والقدرة الحصانية والمسافة المقطوعة التي تعطيها وغيرها الكثير. يعد التنبؤ بسعر السيارة أحد مجالات البحث الرئيسيةفي التعلم الآلي. لذلك إذا كنت تريد معرفة كيفية تدريب نموذج التنبؤ بسعر السيارة، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذكفي جولة حول كيفية تدريب نموذج التنبؤ بسعر السيارة باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بأسعار السيارة مع التعلم الآلي

أحد المجالات الرئيسية للبحثفي التعلم الآلي هو التنبؤ بسعر السيارات. يعتمد على التمويل ومجال التسويق. إنه موضوع بحث رئيسيفي التعلم الآلي لأن سعر السيارة يعتمد على العديد من العوامل. بعض العوامل التي تساهم كثيرًافي سعر السيارة هي:

1. الماركة Brand.
2. النموذج Model.
3. القوة الحصانية Horsepower.
4. عدد الأميال Mileage.
5. ميزات السلامة Safety Features.
6. GPS وغيرها الكثير.

إذا تجاهل المرء العلامة التجارية للسيارة، فإن الشركة المصنعة للسيارة تحدد سعر السيارة بشكل أساسي بناءً على الميزات التي يمكن أن تقدمها للعميل. لاحقًا، قد ترفع العلامة التجارية السعر اعتمادًا على حسن نيتها، ولكن أهم العوامل هي الميزات التي تمنحها السيارة لإضافة قيمة إلى حياتك. لذلك،في القسم أدناه، سوف أطلعك على مهمة تدريب نموذج التنبؤ بسعر السيارة باستخدام التعلم الآلي باستخدام لغة برمجة بايثون.

نموذج التنبؤ بسعر السيارة باستخدام لغة بايثون

تم تنزيل مجموعة البيانات التي أستخدمها هنا لتدريب نموذج التنبؤ بأسعار السيارة من Kaggle. يحتوي على بيانات حول جميع الميزات الرئيسية التي تساهمفي سعر السيارة. فلنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

مجموعة البيانات

```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
```

```
data = pd.read_csv("CarPrice.csv")
data.head()
```

	car_ID	symboling	CarName	...	citympg	highwaympg	price
0	1	3	alfa-romero giulia	...	21	27	13495.0
1	2	3	alfa-romero stelvio	...	21	27	16500.0
2	3	1	alfa-romero Quadrifoglio	...	19	26	16500.0
3	4	2	audi 100 ls	...	24	30	13950.0
4	5	2	audi 100ls	...	18	22	17450.0

[5 rows x 26 columns]

يوجد 26 عمودًافي مجموعة البيانات هذه، لذلك من المهم جدًا التحقق مما إذا كانت مجموعة البيانات هذه تحتوي على قيم فارغة (null values) أم لا قبل المضي قدمًا:

```
data.isnull().sum()
```

```
car_ID            0
symboling         0
CarName           0
fueltype          0
aspiration        0
doornumber        0
carbody           0
drivewheel        0
enginelocation    0
wheelbase         0
carlength         0
carwidth          0
carheight         0
curbweight        0
enginetype        0
cylindernumber    0
enginesize        0
fuelsystem        0
boreratio         0
stroke            0
compressionratio  0
horsepower        0
peakrpm           0
citympg           0
highwaympg        0
price             0
dtype: int64
```

لذلك لا تحتوي مجموعة البيانات هذه على أي قيم فارغة، فلنلقِ الآن نظرة على بعض الأفكار المهمة الأخرى للحصول على فكرة عن نوع البيانات التي نتعامل معها:

```
data.info()
```

```
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   car_ID            205 non-null    int64
 1   symboling         205 non-null    int64
 2   CarName           205 non-null    object
 3   fueltype          205 non-null    object
 4   aspiration        205 non-null    object
 5   doornumber        205 non-null    object
 6   carbody           205 non-null    object
 7   drivewheel        205 non-null    object
 8   enginelocation    205 non-null    object
 9   wheelbase         205 non-null    float64
 10  carlength         205 non-null    float64
 11  carwidth          205 non-null    float64
 12  carheight         205 non-null    float64
 13  curbweight        205 non-null    int64
 14  enginetype        205 non-null    object
 15  cylindernumber    205 non-null    object
 16  enginesize        205 non-null    int64
 17  fuelsystem        205 non-null    object
 18  boreratio         205 non-null    float64
 19  stroke            205 non-null    float64
 20  compressionratio  205 non-null    float64
 21  horsepower        205 non-null    int64
 22  peakrpm           205 non-null    int64
 23  citympg           205 non-null    int64
 24  highwaympg        205 non-null    int64
 25  price             205 non-null    float64
dtypes: float64(8), int64(8), object(10)
memory usage: 41.8+ KB
```

```
print(data.describe())
```

```
            car_ID   symboling   wheelbase  ...     citympg  highwaympg          price
count   205.000000  205.000000  205.000000  ...  205.000000  205.000000    205.000000
mean    103.000000    0.834146   98.756585  ...   25.219512   30.751220  13276.710571
std      59.322565    1.245307    6.021776  ...    6.542142    6.886443   7988.852332
min       1.000000   -2.000000   86.600000  ...   13.000000   16.000000   5118.000000
25%      52.000000    0.000000   94.500000  ...   19.000000   25.000000   7788.000000
50%     103.000000    1.000000   97.000000  ...   24.000000   30.000000  10295.000000
75%     154.000000    2.000000  102.400000  ...   30.000000   34.000000  16503.000000
max     205.000000    3.000000  120.900000  ...   49.000000   54.000000  45400.000000

[8 rows x 16 columns]
```

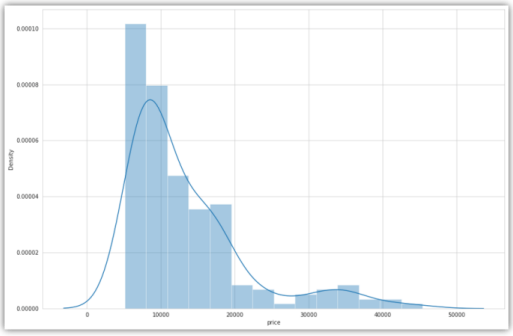
```
data.CarName.unique()
```

```
array(['alfa-romero giulia', 'alfa-romero stelvio',
       'alfa-romero Quadrifoglio', 'audi 100 ls', 'audi 100ls',
       'audi fox', 'audi 5000', 'audi 4000', 'audi 5000s (diesel)',
       'bmw 320i', 'bmw x1', 'bmw x3', 'bmw z4', 'bmw x4', 'bmw x5',
       'chevrolet impala', 'chevrolet monte carlo', 'chevrolet vega 2300',
```

```
'dodge rampage', 'dodge challenger se', 'dodge d200',
'dodge monaco (sw)', 'dodge colt hardtop', 'dodge colt (sw)',
'dodge coronet custom', 'dodge dart custom',
'dodge coronet custom (sw)', 'honda civic', 'honda civic cvcc',
'honda accord cvcc', 'honda accord lx', 'honda civic 1500 gl',
'honda accord', 'honda civic 1300', 'honda prelude',
'honda civic (auto)', 'isuzu MU-X', 'isuzu D-Max ',
'isuzu D-Max V-Cross', 'jaguar xj', 'jaguar xf', 'jaguar xk',
'maxda rx3', 'maxda glc deluxe', 'mazda rx2 coupe', 'mazda rx-4',
'mazda glc deluxe', 'mazda 626', 'mazda glc', 'mazda rx-7 gs',
'mazda glc 4', 'mazda glc custom l', 'mazda glc custom',
'buick electra 225 custom', 'buick century luxus (sw)',
'buick century', 'buick skyhawk', 'buick opel isuzu deluxe',
'buick skylark', 'buick century special',
'buick regal sport coupe (turbo)', 'mercury cougar',
'mitsubishi mirage', 'mitsubishi lancer', 'mitsubishi outlander',
'mitsubishi g4', 'mitsubishi mirage g4', 'mitsubishi montero',
'mitsubishi pajero', 'Nissan versa', 'nissan gt-r', 'nissan rogue',
'nissan latio', 'nissan titan', 'nissan leaf', 'nissan juke',
'nissan note', 'nissan clipper', 'nissan nv200', 'nissan dayz',
'nissan fuga', 'nissan otti', 'nissan teana', 'nissan kicks',
'peugeot 504', 'peugeot 304', 'peugeot 504 (sw)', 'peugeot 604sl',
'peugeot 505s turbo diesel', 'plymouth fury iii',
'plymouth cricket', 'plymouth satellite custom (sw)',
'plymouth fury gran sedan', 'plymouth valiant', 'plymouth duster',
'porsche macan', 'porcshce panamera', 'porsche cayenne',
'porsche boxter', 'renault 12tl', 'renault 5 gtl', 'saab 99e',
'saab 99le', 'saab 99gle', 'subaru', 'subaru dl', 'subaru brz',
'subaru baja', 'subaru r1', 'subaru r2', 'subaru trezia',
'subaru tribeca', 'toyota corona mark ii', 'toyota corona',
'toyota corolla 1200', 'toyota corona hardtop',
'toyota celica gt liftback', 'toyota corolla tercel',
'toyota corona liftback', 'toyota starlet', 'toyota tercel',
'toyota cressida', 'toyota celica gt', 'toyouta tercel',
'vokswagen rabbit', 'volkswagen 1131 deluxe sedan',
'volkswagen model 111', 'volkswagen type 3', 'volkswagen 411 (sw)',
'volkswagen super beetle', 'volkswagen dasher', 'vw dasher',
'vw rabbit', 'volkswagen rabbit', 'volkswagen rabbit custom',
'volvo 145e (sw)', 'volvo 144ea', 'volvo 244dl', 'volvo 245',
'volvo 264gl', 'volvo diesel', 'volvo 246'], dtype=object)
```

من المفترض أن يكون عمود السعرفي مجموعة البيانات هذه هو العمود الذي نحتاج إلى توقع قيمه. دعونا نرى توزيع قيم عمود السعر:

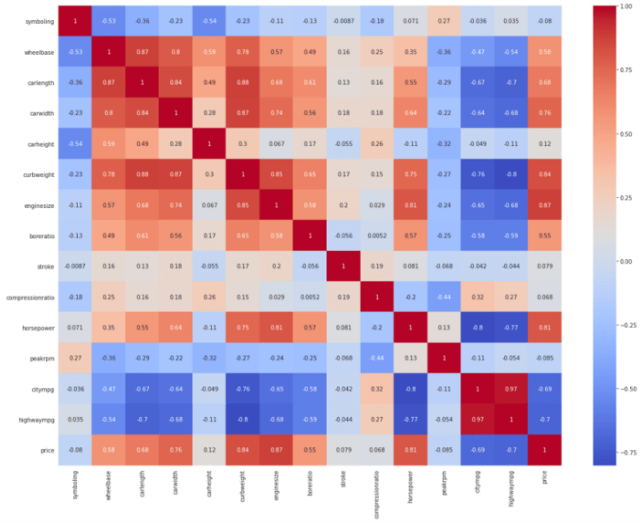
```
sns.set_style("whitegrid")
plt.figure(figsize=(15, 10))
sns.distplot(data.price)
plt.show()
```



دعنا الآن نلقي نظرة على الارتباط بين جميع ميزات مجموعة البيانات هذه:

	car_ID	symboling	...	highwaympg	price
car_ID	1.000000	-0.151621	...	0.011255	-0.109093
symboling	-0.151621	1.000000	...	0.034606	-0.079978
wheelbase	0.129729	-0.531954	...	-0.544082	0.577816
carlength	0.170636	-0.357612	...	-0.704662	0.682920
carwidth	0.052387	-0.232919	...	-0.677218	0.759325
carheight	0.255960	-0.541038	...	-0.107358	0.119336
curbweight	0.071962	-0.227691	...	-0.797465	0.835305
enginesize	-0.033930	-0.105790	...	-0.677470	0.874145
boreratio	0.260064	-0.130051	...	-0.587012	0.553173
stroke	-0.160824	-0.008735	...	-0.043931	0.079443
compressionratio	0.150276	-0.178515	...	0.265201	0.067984
horsepower	-0.015006	0.070873	...	-0.770544	0.808139
peakrpm	-0.203789	0.273606	...	-0.054275	-0.085267
citympg	0.015940	-0.035823	...	0.971337	-0.685751
highwaympg	0.011255	0.034606	...	1.000000	-0.697599
price	-0.109093	-0.079978	...	-0.697599	1.000000

[16 rows x 16 columns]




```
plt.figure(figsize=(20, 15))
correlations = data.corr()
```

```
sns.heatmap(correlations, cmap="coolwarm", annot=True)
plt.show()
```

تدريب نموذج التنبؤ بسعر السيارة

سأستخدم خوارزمية انحدار شجرة القرار decision tree لتدريب نموذج التنبؤ بسعر السيارة. لذلك دعونا نقسم البيانات إلى مجموعات تدريب واختبار ونستخدم خوارزمية انحدار شجرة القرار لتدريب النموذج:

```
predict = "price"
data = data[["symboling", "wheelbase", "carlength",
             "carwidth", "carheight", "curbweight",
             "enginesize", "boreratio", "stroke",
             "compressionratio", "horsepower", "peakrpm",
             "citympg", "highwaympg", "price"]]
x = np.array(data.drop([predict], 1))
y = np.array(data[predict])

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2)

from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor()
model.fit(xtrain, ytrain)
predictions = model.predict(xtest)

from sklearn.metrics import mean_absolute_error
model.score(xtest, predictions)
```

```
1.0
```

يعطي النموذج دقة 100٪في مجموعة الاختبار، وهو أمر ممتاز.

الملخص

هذه هي الطريقة التي يمكنك بها تدريب نموذج التعلم الآلي لمهمة التنبؤ بأسعار السيارات باستخدام لغة برمجة بايثون. إنه موضوع بحث رئيسيفي التعلم الآلي لأن سعر السيارة يعتمد على العديد من العوامل. أتمنى أن تكون قد أحببت هذا المقال حول مهمة تدريب نموذج للتنبؤ بأسعار السيارات باستخدام التعلم الآلي.

30) اكتشاف البريد العشوائي باستخدام التعلم الآلي Spam Detection with Machine Learning

يعد اكتشاف البريد غير المرغوب فيه (Spam Detection)في رسائل البريد الإلكتروني والرسائل أحد التطبيقات الرئيسية التي تحاول كل شركة تقنية كبيرة تحسينها لعملائها. يعد تطبيق المراسلة الرسمي من Apple وGmail من Google مثالين رائعين لمثل هذه التطبيقات حيث يعمل اكتشاف الرسائل غير المرغوب فيها بشكل جيد لحماية المستخدمين من تنبيهات الرسائل غير المرغوب فيها. لذلك، إذا كنت تتطلع إلى إنشاء نظام للكشف عن الرسائل غير المرغوب فيها، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة اكتشاف البريد العشوائي باستخدام التعلم الآلي باستخدام بايثون.

اكتشاف البريد العشوائي

عندما ترسل تفاصيل حول بريدك الإلكتروني أو رقم الاتصال الخاص بك على أي منصة، أصبح من السهل على تلك المنصات تسويق منتجاتها عن طريق الإعلان عنها عن طريق إرسال رسائل بريد إلكتروني أو عن طريق إرسال رسائل مباشرة إلى رقم الاتصال الخاص بك. ينتج عن هذا الكثير من التنبيهات والإشعارات بشأن البريد العشوائيفي صندوق الوارد الخاص بك. هذا هو المكان الذي تأتي فيه مهمة اكتشاف البريد العشوائي.

يعني Spam detection اكتشاف الرسائل غير المرغوب فيها أو رسائل البريد الإلكتروني من خلال فهم محتوى النص بحيث يمكنك فقط تلقي إشعارات حول الرسائل أو رسائل البريد الإلكتروني المهمة جدًا بالنسبة لك. إذا تم العثور على رسائل غير مرغوب فيها، فسيتم نقلها تلقائيًا إلى مجلد البريد العشوائي ولن يتم إخطارك بهذه التنبيهات. يساعد هذافي تحسين تجربة المستخدم، حيث يمكن للعديد من تنبيهات البريد العشوائي أن تزعج العديد من المستخدمين.

اكتشاف البريد العشوائي باستخدام بايثون

آمل أن تفهم الآن ما هو اكتشاف الرسائل غير المرغوب فيها، فلنرى الآن كيفية تدريب نموذج التعلم الآلي على اكتشاف تنبيهات الرسائل غير المرغوب فيها باستخدام بايثون. سأبدأ هذه المهمة عن طريق استيراد مكتبات بايثون الضرورية ومجموعة البيانات التي تحتاجها لهذه المهمة:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
data = pd.read_csv("https://raw.githubusercontent.com/amankharwal/SMS-
Spam-Detection/master/spam.csv", encoding= 'latin-1')
data.head()
```

	class	message	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

من مجموعة البيانات هذه، الفئة class والرسالة message هما الميزتان الوحيدتان اللتان نحتاجهما لتدريب نموذج التعلم الآلي على اكتشاف الرسائل غير المرغوب فيها، لذلك دعونا نحدد هذين العمودين كمجموعة بيانات جديدة:

```
data = data[["class", "message"]]
```

دعنا الآن نقسم مجموعة البيانات هذه إلى مجموعات تدريب واختبار وتدريب النموذج على اكتشاف الرسائل غير المرغوب فيها:

```
x = np.array(data["message"])
y = np.array(data["class"])
cv = CountVectorizer()
X = cv.fit_transform(x) # Fit the Data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

clf = MultinomialNB()
clf.fit(X_train,y_train)
```

دعنا الآن نختبر هذا النموذج من خلال أخذ إدخال المستخدم كرسالة لاكتشاف ما إذا كان بريداً عشوائيًا أم لا:

```
sample = input('Enter a message:')
data = cv.transform([sample]).toarray()
print(clf.predict(data))
```

```
Enter a message:You won $40 cash price
['spam']
```

الملخص

هذه هي الطريقة التي يمكنك بها تدريب نموذج التعلم الآلي على مهمة اكتشاف ما إذا كانت رسالة بريد إلكتروني أو رسالة بريد عشوائي أم لا. يكتشف كاشف الرسائل غير المرغوب فيها رسائل البريد العشوائي أو رسائل البريد الإلكتروني من خلال فهم محتوى النص بحيث يمكنك فقط تلقي إشعارات حول الرسائل أو رسائل البريد الإلكتروني المهمة جدًا بالنسبة لك. آمل أن تكون قد أحببت هذه المقالة حول مهمة الكشف عن تنبيهات البريد العشوائي باستخدام التعلم الآلي باستخدام بايثون.

31) التنبؤ بدرجات الطلاب باستخدام التعلم الآلي Student Grades Prediction with Machine Learning

في الدراسات العليا، يجد العديد من الطلاب صعوبةفي تحقيق درجات جيدة لأنهم لا يحصلون على دعم كبيرفي دورات التعليم العالي مقارنة بالدعم الذي يتلقاه الطلابفي المدارس. يمكننا استخدام التعلم الآلي لمهمة التنبؤ بدرجات الطالب حتى يتمكن المدرسون من مساعدة الطلاب على الاستعداد للموضوعات التي تم توقع انخفاض درجات الطلاب فيها.في هذه المقالة، سوف أطلعك على مهمة التنبؤ بدرجات الطلاب باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بدرجات الطالب

الجامعات هي أماكن مرموقة للغاية للوصول إلى التعليم العالي. لكن مقدار الرسوم التي تفرضها الجامعات اليوم لا يساوي أبدًا الدعم الذي تقدمه للطلاب. يحتاج بعض الطلاب إلى الكثير من الاهتمام من المعلمين لأنه إذا لم يتم إيلاء اهتمام خاص لأولئك الطلاب الذين لم يحصلوا على درجات جيدة، فقد يكون ذلك ضارًا بحالتهم العاطفية وحياتهم المهنية على المدى الطويل.

باستخدام خوارزميات التعلم الآلي، يمكننا التنبؤ بمدى جودة أداء الطلاب حتى نتمكن من مساعدة الطلاب الذين من المتوقع أن تكون درجاتهم منخفضة. يعتمد توقع درجات الطلاب على مشكلة الانحدارفي التعلم الآلي.في القسم أدناه، سوف آخذك خلال مهمة التنبؤ بدرجات الطلاب باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بدرجات الطالب باستخدام لغة بايثون

آمل أن تكون قد فهمت الآن سبب حاجتنا للتنبؤ بدرجات الطالب. دعنا الآن نرى كيف يمكننا استخدام التعلم الآلي لمهمة التنبؤ بدرجات الطلاب باستخدام بايثون. سأبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.utils import shuffle

data = pd.read csv("student-mat.csv")
data.head()
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother
1	GP	F	17	U	GT3	T	1	1	at_home	other	course	father
2	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother
3	GP	F	15	U	GT3	T	4	2	health	services	home	mother
4	GP	F	16	U	GT3	T	3	3	other	other	home	father

تستند مجموعة البيانات التي أستخدمها لمهمة التنبؤ بدرجات الطلاب إلى إنجازات طلاب المدارس البرتغالية.في مجموعة البيانات هذه، يمثل G1 درجات الفترة الأولى، ويمثل G2 درجات الفترة الثانية، ويمثل G3 الدرجات النهائية. الآن دعنا نجهز البيانات ودعنا نرى كيف يمكننا توقع الدرجات النهائية للطلاب:

```
data = data[["G1", "G2", "G3", "studytime", "failures", "absences"]]
predict = "G3"
x = np.array(data.drop([predict], 1))
y = np.array(data[predict])

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2)
```

في الكود أعلاه، قمت أولاً بتحديد الأعمدة الضرورية التي نحتاجها لتدريب نموذج التعلم الآلي لمهمة التنبؤ بدرجات الطلاب. ثم أعلنت أن عمود G3 هو التسمية المستهدفة لدينا، ثم قسمت مجموعة البيانات إلى اختبار بنسبة 20٪ وتدريب بنسبة 80٪. دعنا الآن نرى كيفية تدريب نموذج الانحدار الخطي linear regression لمهمة التنبؤ بدرجات الطلاب:

```
linear_regression = LinearRegression()
linear_regression.fit(xtrain, ytrain)
accuracy = linear_regression.score(xtest, ytest)
print(accuracy)
```

```
0.8432876775479776
```

أعطى نموذج الانحدار الخطي دقة حوالي 84٪ وهي ليست سيئةفي هذه المهمة. الآن دعنا نلقي نظرة على التنبؤات التي قدمها نموذج التنبؤ بالدرجات للطلاب:

```
predictions = linear_regression.predict(xtest)
for i in range(len(predictions)):
    print(predictions[x], xtest[x], [ytest[x]])
```

```
[[16.16395534 14.23423176 14.08532841  5.28096434 14.23423176]
 [16.16395534 16.16395534 14.08532841  5.28096434  7.97291422]
 [14.52779998 11.92149651 14.08532841  9.13993948  4.71694746]
 ...
 [ 4.71694746 11.92149651  3.9451298   9.13993948  9.13993948]
 [12.56424351  4.92497623  3.9451298   5.28096434  5.28096434]
 [11.92149651  9.05247158  3.9451298   5.28096434 16.16395534]] [[[15 16  2  0  2]
  [15 14  2  0  2]
  [15 14  3  0  6]
  [ 7  6  2  0 10]
  [15 14  2  0  2]]....
```

الملخص

هذه هي الطريقة التي يمكنك بها تدريب نموذج الانحدار الخطي لمهمة تنبؤ درجات الطلاب باستخدام التعلم الآلي باستخدام بايثون. يمكنك القيام بالكثير باستخدام مجموعة البيانات هذه، ويمكنك العثور على المعلومات الكاملة حول مجموعة البيانات هذه من هنا.

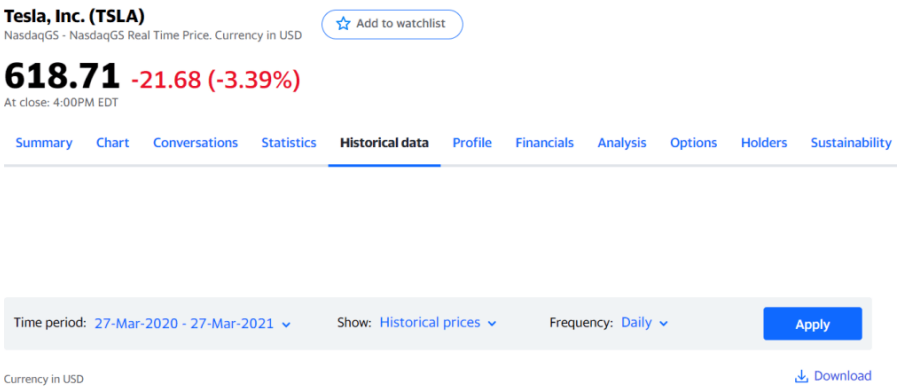
32) التنبؤ بسعر سهم Tesla مع التعلم الآلي Tesla Stock Price Prediction with Machine Learning

Tesla هي شركة أمريكية للسيارات تهدف إلى تسريع انتقال العالم نحو الطاقة المستدامة. قبل أيام قليلة من ارتفاع أسعار أسهم Tesla، جعل Elon Musk أغنى شخصفي العالم. يُعد التنبؤ بأسعار الأسهم حالة استخدام رائعة للتعلم الآلي، لذلكفي هذه المقالة، سوف آخذك خلال مهمة توقع أسعار أسهم Tesla مع التعلم الآلي باستخدام بايثون.

التنبؤ بسعر سهم Tesla مع التعلم الآلي

يعد التنبؤ بأسعار الأسهم حالة استخدام رائعة للتعلم الآلي لكل من تحليل السلاسل المالية والزمنية. كانت Teslaفي عيون العالم لفترة طويلة الآن حيث تدعم حكومات العديد من البلدان في جميع أنحاء العالم رؤية Tesla. لذلكفي هذه المقالة، سوف آخذك من خلال برنامج تعليمي حول كيفية استخدام نموذج Facebook Prophet لمهمة التنبؤ بسعر سهم Tesla.

تم تنزيل مجموعة البيانات التي سأستخدمها هنا من موقع yahoo finance. لتنزيل مجموعة البيانات هذه، ما عليك سوى زيارة موقع yahoo finance وابحث عن TSLA. سترى الداشبورد كما هو موضحفي الصورة أدناه:



هنا عليك النقر فوق الأسعار التاريخية ثم النقر فوق تنزيل. سيتم تنزيل مجموعة البيانات باسم "TSLA.csv".

التنبؤ بسعر سهم Tesla باستخدام لغة بايثون

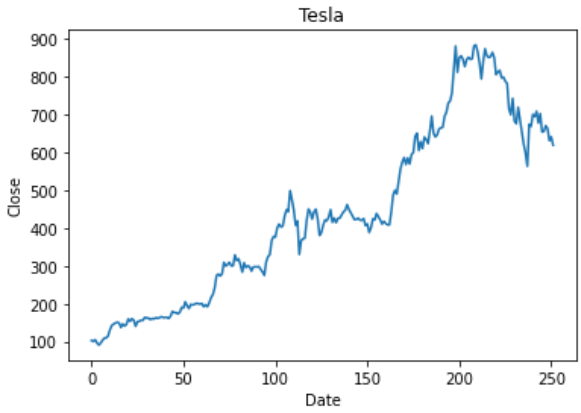
أتمنى أن تكون قد قمت بتنزيل البيانات التاريخية لأسعار أسهم Tesla بسهولة باتباع الخطوات المذكورةفي القسم أعلاه. دعنا الآن نرى كيفية توقع أسعار أسهم Tesla باستخدام التعلم الآلي باستخدام بايثون. سأبدأ هنا باستيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from fbprophet import Prophet

data = pd.read_csv("TSLA.csv")
data.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2020-03-27	101.000000	105.160004	98.806000	102.872002	102.872002	71887000
1	2020-03-30	102.052002	103.330002	98.246002	100.426003	100.426003	59990500
2	2020-03-31	100.250000	108.592003	99.400002	104.800003	104.800003	88857500
3	2020-04-01	100.800003	102.790001	95.019997	96.311996	96.311996	66766000
4	2020-04-02	96.206001	98.851997	89.279999	90.893997	90.893997	99292000

قبل المضي قدمًا، دعنا نتخيل عمود الإغلاق "Close "في مجموعة البيانات والذي يمثل أسعار الإغلاق لكل يوم:

```
close = data['Close']
ax = close.plot(title='Tesla')
ax.set_xlabel('Date')
ax.set_ylabel('Close')
plt.show()
```



نحتاج فقط إلى عمودين من مجموعة البيانات هذه (التاريخ والإغلاق)، لذا فلنقم بإنشاء إطار بيانات جديد بهذين العمودين فقط:

```
data["Date"] = pd.to_datetime(data["Date"],
infer_datetime_format=True)
data = data[["Date", "Close"]]
```

نظرًا لأننا نستخدم نموذج Facebook prophet هنا للتنبؤ بأسعار أسهم Tesla ، فنحن بحاجة إلى إعادة تسمية الأعمدة:

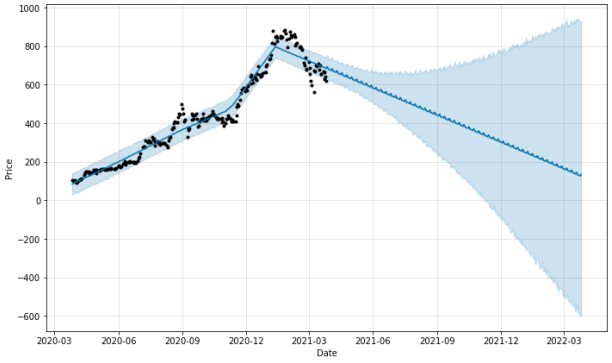
```
data = data.rename(columns={"Date" : "ds", "Close" : "y" })
```

لذلك قمنا بإعداد مجموعة البيانات لنموذج Facebook prophet، والآن دعونا نتنبأ بأسعار أسهم Tesla:

```
model = Prophet()
model.fit(data)
predict = model.make_future_dataframe(periods=365)
forcast = model.predict(predict)
forcast[["ds", "yhat", "yhat_lower", "yhat_upper"]].tail()
```

	ds	yhat	yhat_lower	yhat_upper
612	2022-03-22	131.630090	-585.268250	931.611531
613	2022-03-23	129.584278	-583.545636	935.285420
614	2022-03-24	127.202999	-593.418911	930.525966
615	2022-03-25	125.492270	-601.302243	924.233573
616	2022-03-26	134.596307	-595.884292	930.702026

```
graph = model.plot(forcast, xlabel="Date", ylabel="Price")
```



الملخص

يبدو أن أسعار أسهم Tesla ستنخفضفي المستقبل القريب إذا لم يأتوا بفكرة جديدة تمثل رؤيتهم. قد يكون هذا ممكنًا لأن الشركات الأخرى بدأت أيضًافي تصنيع السيارات الكهربائية بسعر منخفض جدًا مقارنةً بـ Tesla .آمل أن تكون قد أحببت هذه المقالة حول مهمة التنبؤ بأسعار أسهم Tesla مع التعلم الآلي باستخدام بايثون.

33) التنبؤ بمعدل النقر للظهور من خلال التعلم الآلي Click-Through Rate Prediction with Machine Learning

في أي وكالة إعلانية، من المهم جداً توقع المستخدمين الأكثر ربحية والذين من المرجح جداً أن يستجيبوا للإعلانات المستهدفة.في هذه المقالة، سوف أطلعك على كيفية تدريب نموذج لمهمة التنبؤ بمعدل النقر (click-through rate prediction) باستخدام التعلم الآلي باستخدام بايثون.

نموذج التنبؤ بمعدل النقر مع التعلم الآلي

من خلال توقع نسبة النقر إلى الظهور، تختار شركة إعلانات الزائرين المحتملين الذين من المرجح أن يستجيبوا للإعلانات، وتحلل سجل التصفح الخاص بهم وتعرض الإعلانات الأكثر صلة بناءً على اهتمام المستخدم.

هذه المهمة مهمة لكل وكالة إعلانية لأن القيمة التجارية للترويج على الإنترنت تعتمد فقط على كيفية استجابة المستخدم لها. تُعد استجابة المستخدم للإعلانات ذات قيمة كبيرة لكل شركة إعلانية لأنها تتيح للشركة تحديد الإعلانات الأكثر صلة بالمستخدمين.

في القسم أدناه، سوف آخذك إلى كيفية إنشاء نموذج لمهمة التنبؤ بمعدل النقر باستخدام التعلم الآلي باستخدام بايثون. هذه المهمة مهمة جداً لأولئك الذين يرغبونفي العمل كعالم بياناتفي أي شركة تتعامل مع إعلانات مثل Google وFacebook.

نموذج التنبؤ بمعدل النقر باستخدام بايثون

لنبدأ الآنفي مهمة نموذج للتنبؤ بمعدل النقر إلى الظهور باستخدام التعلم الآلي عن طريق استيراد مجموعة البيانات:

```
import pandas as pd
data = pd.read_csv('advertising.csv')
print(data.head())
```

```
   Daily Time Spent on Site  Age  ...            Timestamp  Clicked on Ad
0                     68.95   35  ...  2016-03-27 00:53:11              0
1                     80.23   31  ...  2016-04-04 01:39:02              0
2                     69.47   26  ...  2016-03-13 20:35:42              0
3                     74.15   29  ...  2016-01-10 02:31:19              0
4                     68.37   35  ...  2016-06-03 03:36:18              0

[5 rows x 10 columns]
```

دعنا الآن نلقي نظرة على البيانات لمعرفة ما إذا كان لدينا أي قيم فارغةفي مجموعة البيانات:

```
print(data.isnull().sum())
```

```
Daily Time Spent on Site    0
Age                         0
Area Income                 0
Daily Internet Usage        0
Ad Topic Line               0
City                        0
Male                        0
Country                     0
Timestamp                   0
Clicked on Ad               0
dtype: int64
```

قبل المضي قدمًا، دعنا نلقي نظرة على أسماء جميع الأعمدةفي مجموعة البيانات:

```
print(data.columns)
```

```
Index(['Daily Time Spent on Site', 'Age', 'Area Income',
       'Daily Internet Usage', 'Ad Topic Line', 'City', 'Male', 'Country',
       'Timestamp', 'Clicked on Ad'],
      dtype='object')
```

دعنا الآن نحضر البيانات حتى نتمكن من التوافق بسهولة مع نموذج التعلم الآلي. سنقوم هنا بإسقاط بعض الأعمدة غير الضرورية:

```
x=data.iloc[0:7,:]
x=x.drop(['Ad Topic Line','City'],axis=1)
x
```

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Male
0	68.95	35	61833.90	256.09	0
1	80.23	31	68441.85	193.77	1
2	69.47	26	59785.94	236.50	0
3	74.15	29	54806.18	245.89	1
4	68.37	35	73889.99	225.58	0
...	...	...	...	...	...
995	72.97	30	71384.57	208.58	1
996	51.30	45	67782.17	134.42	1
997	51.63	51	42415.72	120.37	1
998	55.55	19	41920.79	187.95	0
999	45.01	26	29875.80	178.35	0

1000 rows × 5 columns

```
y=data.iloc[:,9]
y
```

```
0      0
1      0
2      0
3      0
4      0
      ..
995    1
996    1
997    1
998    0
999    1
Name: Clicked on Ad, Length: 1000, dtype: int64
```

الآن سأقسم البيانات إلى مجموعات تدريب واختبار. سأستخدم هنا 70في المائة من البيانات كتدريب و30في المائة كاختبار:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0
.3,random_state=4)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(700, 5)
(700,)
(300, 5)
(300,)
```

نموذج الانحدار اللوجستي:

سأستخدم الآن نموذج الانحدار اللوجستي (logistic regression) للتنبؤ بنسبة النقر إلى الظهور للمستخدمين:

```
from sklearn.linear_model import LogisticRegression
Lr=LogisticRegression(C=0.01,random_state=0)
Lr.fit(x_train,y_train)
y_pred=Lr.predict(x_test)
print(y_pred)
```

```
[0 0 0 1 1 0 0 1 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1
 0 1 0 0 1 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 0 0 1 1 1 1 0 1 0 0 0 0 1 1 0 0 0
 0 1 1 1 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0
 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 1 0 0
 1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 1 1 1 1 0 1 1 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 1 0
 0 1 0 1 0 1 1 1 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 0
 0 1 0 0 0 1 1 0 0 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1
 1 0 1 0 1 0 0 1 0 0 0 1 1 1 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 1 0 1 1 1 1 0 0 1
 1 0 0 0]
```

```
y_pred_proba=Lr.predict_proba(x_test)
print(y_pred_proba)
```

```
[[9.56377730e-01 4.36222698e-02]
 [4.57913871e-02 9.54208613e-01]
 [9.90680226e-01 9.31977394e-03]
 [4.52921708e-03 9.95470783e-01]
 [9.53422807e-01 4.65771934e-02]
 [4.93413634e-01 5.06586366e-01]
 [9.27749026e-01 7.22509737e-02]
 [9.88127401e-01 1.18725991e-02]
 [8.38452014e-01 1.61547986e-01]
 [8.36925286e-01 1.63074714e-01]
 [9.80069836e-01 1.99301643e-02]
 [9.65651589e-01 3.43484110e-02]
 [6.84448458e-02 9.31555154e-01]
 [9.87522934e-01 1.24770659e-02]
 [9.17172887e-01 8.28271130e-02]
 [2.43751163e-01 7.56248837e-01]
 [9.37078134e-01 6.29218662e-02]
 [2.82792965e-02 9.71720703e-01]
 [2.72748965e-01 7.27251035e-01]
 [3.00015981e-02 9.69998402e-01]
 [9.85498737e-01 1.45012626e-02]
 [9.82428592e-01 1.75714076e-02]
 [3.70265237e-03 9.96297348e-01]
 [7.99116022e-02 9.20088398e-01]
 [9.09972207e-01 9.00277931e-02]
 [9.44866412e-01 5.51335881e-02]
 [9.84521236e-01 1.54787639e-02]]
```

الآن دعنا نلقي نظرة على دقة النموذج:

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,y_pred))
```

```
0.8733333333333333
```

لذلك لدينا دقة تبلغ حوالي 87في المائة وهي ليست سيئة لهذا النوع من المشاكل. أخيرًا، دعونا نلقي نظرة على نتيجة f1 score:

```
from sklearn.metrics import f1_score
print(f1_score(y_test,y_pred))
```

```
0.8652482269503545
```

آمل أن تكون قد أحببت هذه المقالة حول نموذج التنبؤ بنسبة النقر إلى الظهور مع التعلم الآلي باستخدام بايثون.

34) اكتشاف السخرية مع التعلم الآلي Sarcasm Detection with Machine Learning

كانت السخرية Sarcasm جزءًا من لغتنا لسنوات عديدة. هذا يعني أن تكون عكس ما تعنيه، عادةً بنبرة صوت مميزة بطريقة ممتعة. إذا كنت تعتقد أن أي شخص يمكنه فهم السخرية، فأنت مخطئ، لأن فهم السخرية يعتمد على مهاراتك اللغوية ومعرفتك بعقول الآخرين. لكن ماذا عن الكمبيوتر؟ هل من الممكن تدريب نموذج للتعلم الآلي يمكنه اكتشاف ما إذا كانت الجملة ساخرة أم لا؟ نعم إنه كذلك! لذلك إذا كنت تريد معرفة كيفية اكتشاف السخرية Sarcasm detection باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة اكتشاف السخرية باستخدام التعلم الآلي باستخدام بايثون.

اكتشاف السخرية مع التعلم الآلي

السخرية تعني أن تكون مضحكا بأن تكون عكس ما تعنيه. لقد كانت جزءًا من كل لغة بشرية لسنوات. اليوم، يتم استخدامه أيضًافي عناوين الأخبار والعديد من منصات التواصل الاجتماعي الأخرى لجذب المزيد من الاهتمام. اكتشاف السخرية هو معالجة لغة طبيعية ومهمة تصنيف ثنائي. يمكننا تدريب نموذج التعلم الآلي لاكتشاف ما إذا كانت الجملة ساخرة أم لا باستخدام مجموعة بيانات من الجمل الساخرة وغير الساخرة التي وجدتهافي Kaggle.

أتمنى أن تكون قد فهمت الآن ما هي السخرية.في القسم أدناه، سأوجهك خلال مهمة اكتشاف السخرية من خلال التعلم الآلي باستخدام لغة برمجة بايثون. يمكن تنزيل مجموعة البيانات التي أستخدمها لهذه المهمة من هنا.

اكتشاف السخرية باستخدام بايثون

لنبدأ الآن بمهمة اكتشاف السخرية باستخدام التعلم الآلي باستخدام بايثون. سأبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import BernoulliNB

data = pd.read_json("Sarcasm.json", lines=True)
print(data.head())
```

```
                                article_link  ... is_sarcastic
0  https://www.huffingtonpost.com/entry/versace-b...  ...            0
1  https://www.huffingtonpost.com/entry/roseanne-...  ...            0
2  https://local.theonion.com/mom-starting-to-fea...  ...            1
3  https://politics.theonion.com/boehner-just-wan...  ...            1
4  https://www.huffingtonpost.com/entry/jk-rowlin...  ...            0

[5 rows x 3 columns]
```

يحتوي عمود "is_sarcastic"في مجموعة البيانات هذه على التسميات التي يتعين علينا توقعها لمهمة اكتشاف السخرية. يحتوي على قيم ثنائية مثل 1 و0، حيث 1 تعني ساخرة sarcastic و 0 ليست ساخرة not sarcastic. من أجل التبسيط، سأحول قيم هذا العمود إلى "sarcastic" و "not sarcastic" بدلاً من 1 و0:

```
data["is_sarcastic"] = data["is_sarcastic"].map({0: "Not Sarcasm", 1:
"Sarcasm"})
print(data.head())
```

```
                                article_link  ... is_sarcastic
0  https://www.huffingtonpost.com/entry/versace-b...  ...  Not Sarcasm
1  https://www.huffingtonpost.com/entry/roseanne-...  ...  Not Sarcasm
2  https://local.theonion.com/mom-starting-to-fea...  ...      Sarcasm
3  https://politics.theonion.com/boehner-just-wan...  ...      Sarcasm
4  https://www.huffingtonpost.com/entry/jk-rowlin...  ...  Not Sarcasm

[5 rows x 3 columns]
```

دعنا الآن نجهز البيانات لتدريب نموذج التعلم الآلي. تحتوي مجموعة البيانات هذه على ثلاثة أعمدة، نحتاج منها فقط إلى عمود "headline" كميزة والعمود "is_sarcastic" كتسمية. لذلك دعونا نحدد هذه الأعمدة ونقسم البيانات إلى مجموعة اختبار 20٪ ومجموعة تدريب 80٪:

```
data = data[["headline", "is_sarcastic"]]
x = np.array(data["headline"])
y = np.array(data["is_sarcastic"])

cv = CountVectorizer()
X = cv.fit_transform(x) # Fit the Data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)
```

سأستخدم الآن خوارزمية Bernoulli Naive Bayes لتدريب نموذج لمهمة اكتشاف السخرية:

```
model = BernoulliNB()
model.fit(X_train, y_train)
print(model.score(X_test, y_test))
```

```
0.8448146761512542
```

دعنا الآن نستخدم نصًا ساخرًا كمدخلات لاختبار ما إذا كان نموذج التعلم الآلي لدينا يكتشف السخرية أم لا:

```
user = input("Enter a Text: ")
data = cv.transform([user]).toarray()
output = model.predict(data)
print(output)
```

```
Enter a Text: Cows lose their jobs as milk prices drop
['Sarcasm']
```

الملخص

هذه هي الطريقة التي يمكنك بها استخدام التعلم الآلي لاكتشاف السخرية باستخدام لغة برمجة بايثون. كانت السخرية جزءًا من لغتنا لسنوات عديدة. هذا يعني أن تكون عكس ما تعنيه، عادةً بنبرة صوت مميزة بطريقة ممتعة. أتمنى أن تكون قد أحببت هذه المقالة حول مهمة اكتشاف السخرية باستخدام التعلم الآلي باستخدام بايثون.

35) التنبؤ بمتابعي وسائل التواصل الاجتماعي باستخدام التعلم الآلي Social Media Followers Prediction with Machine Learning

يوجد اليوم العديد من منصات الوسائط الاجتماعية حيث ستجد العديد من منشئي المحتوىفي العديد من أنواع المجالات. بصفتك مستهلكًا لوسائل التواصل الاجتماعي، قد لا يكون عدد المتابعين لديك محل اهتمامك، ولكن بصفتك منشئ محتوى أو كرجل أعمال، فإن عدد المتابعين لديك مهم للوصول إلى المحتوى الخاص بك لمزيد من الجمهور. لذا، فإن مهمة التنبؤ بمتابعي وسائل التواصل الاجتماعي مهمة جدًا لكل منشئ محتوى وكل عمل يعتمد على وسائل التواصل الاجتماعي. لذلك إذا كنت تريد معرفة كيفية التنبؤ بمتابعيك على وسائل التواصل الاجتماعي للشهر المقبل، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة التنبؤ متابعي وسائل التواصل الاجتماعي مع التعلم الآلي باستخدام بايثون.

التنبؤ بمتابعي وسائل التواصل الاجتماعي

للتنبؤ بالزيادةفي عدد المتابعين الذين يمكن أن تتوقع رؤيتهم، فأنت بحاجة إلى مجموعة بيانات لمتابعيك على وسائل التواصل الاجتماعي والتي يمكن أن تعرض لك أنشطة الأشخاصفي حسابك على وسائل التواصل الاجتماعي مثل:

1. كم عدد الأشخاص الذين تابعوك كل شهر.
2. عدد المشاهدات الناتجة عن عدد المتابعين.
3. كم من متابعيك يلغون متابعتك كل شهر.

لذلك من الصعب جدًا العثور على مجموعة البيانات هذه بين أكثر منصات التواصل الاجتماعي شيوعًا مثل Facebook وInstagram لأن هذه المنصات لا توفر أي بيانات متعلقة بمتابعيك. لذلك بالنسبة لمهمة توقع متابعي وسائل التواصل الاجتماعي مع التعلم الآلي، قمت بجمع البيانات من حساب الوسائط الاجتماعية الخاص بي على Medium، وهو عبارة عن منصة وسائط اجتماعية لكتاب المحتوى والمدونين والباحثين. يمكنك استخدام نفس العمليةفي مجموعة البيانات الخاصة بك سواء حصلت عليها من Medium أو Instagram أو أي تطبيق وسائط اجتماعية آخر للتنبؤ بمتابعيك على وسائل التواصل الاجتماعي. للممارسة، يمكنك استخدام نفس مجموعة البيانات التي أستخدمها.

التنبؤ بمتابعي الوسائط الاجتماعية باستخدام بايثون

سأبدأ مهمة التنبؤ بمتابعي وسائل التواصل الاجتماعي باستخدام التعلم الآلي عن طريق استيراد مكتبات بايثون الضرورية ومجموعة البيانات التي جمعتها عن متابعيني من Medium:

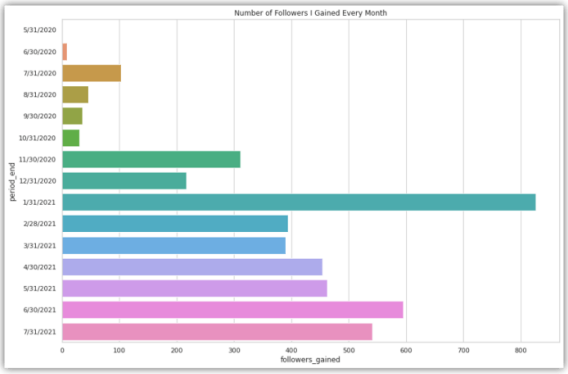
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

data = pd.read_csv("stats.csv")
data.drop(data.tail(1).index, inplace=True)
data.head()
```

	period_start	period_end	...	subscribers_total	views
0	5/1/2020	5/31/2020	...	0	128.0
1	6/1/2020	6/30/2020	...	0	16130.0
2	7/1/2020	7/31/2020	...	0	14616.0
3	8/1/2020	8/31/2020	...	0	4053.0
4	9/1/2020	9/30/2020	...	0	5153.0

[5 rows x 11 columns]

في السطر السابع من الكود أعلاه، قمت بحذف الصف الأخير من مجموعة البيانات لأنه يحتوي على بيانات حول هذا الشهر. الآن سألقي نظرة على عدد المتابعين الذين اكتسبتهم كل شهر على حسابي منذ أن انضممت إلى منصة التواصل الاجتماعي هذه:

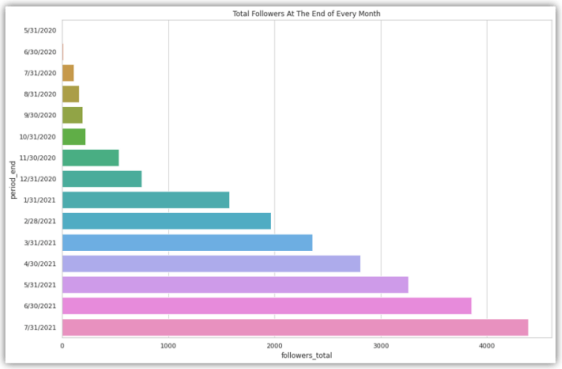
```
plt.figure(figsize=(15, 10))
sns.set_theme(style="whitegrid")
plt.title("Number of Followers I Gained Every Month")
sns.barplot(x="followers_gained", y="period_end", data=data)
plt.show()
```



الآن دعنا نلقي نظرة على العدد الإجمالي للمتابعين الذين ينتهي بي الأمر معهم كل شهر:

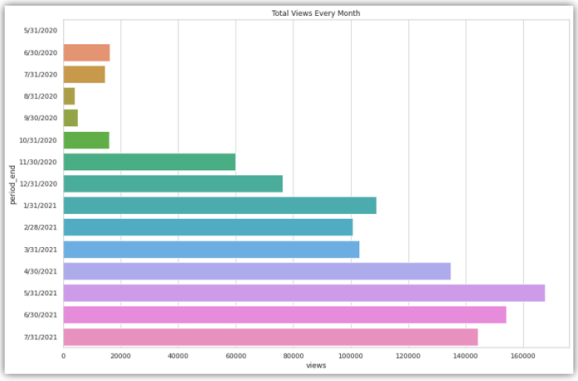
```
plt.figure(figsize=(15, 10))
sns.set_theme(style="whitegrid")
plt.title("Total Followers At The End of Every Month")
```

```
sns.barplot(x="followers_total", y="period_end", data=data)
plt.show()
```



دعنا الآن نلقي نظرة على إحدى أهم الميزات، وهي إجمالي عدد المشاهدات التي أحصل عليها كل شهر:

```
plt.figure(figsize=(15, 10))
sns.set_theme(style="whitegrid")
plt.title("Total Views Every Month")
sns.barplot(x="views", y="period_end", data=data)
plt.show()
```



سأستخدم الآن مكتبة autots في بايثون، والتي تعد واحدة من أفضل مكتبات علوم البيانات للتنبؤ بالسلسلة الزمنية (time series forecasting). إذا لم تكن قد استخدمت هذه المكتبة من قبل، فيمكنك تثبيتها بسهولة على نظامك باستخدام الأمر pip:

```
pip install autots
```

الآن إليك كيف يمكننا توقع الزيادةفي عدد المتابعين الذين نتوقع رؤيتهم خلال الأشهر الأربعة المقبلة:

```
from autots import AutoTS
model = AutoTS(forecast_length=4, frequency='infer',
ensemble='simple')
```

```
model = model.fit(data, date_col='period_end',
value_col='followers_gained', id_col=None)
prediction = model.predict()
forecast = prediction.forecast
print(forecast)
```

```
            followers_gained
2021-08-31        693.465876
2021-09-30        617.750000
2021-10-31        650.000000
2021-11-30        634.750000
```

الملخص

إذن هذه هي الطريقة التي يمكنك بها توقع الزيادةفي عدد متابعيك على أي منصة وسائط اجتماعية. بصفتك مستهلكًا لوسائل التواصل الاجتماعي، قد لا يكون عدد المتابعين لديك محل اهتمامك، ولكن بصفتك منشئ محتوى أو كرجل أعمال، فإن عدد المتابعين لديك مهم للوصول إلى المحتوى الخاص بك لمزيد من الجمهور. آمل أن تكون قد أحببت هذه المقالة حول مهمة التنبؤ بمتابعي وسائل التواصل الاجتماعي مع التعلم الآلي باستخدام بايثون.

36) اكتشاف خطاب الكراهية باستخدام التعلم الآلي Hate Speech Detection with Machine Learning

يعتبر خطاب الكراهية (Hate speech) من القضايا الخطيرة التي نراها يوميًا على منصات التواصل الاجتماعي مثل Twitter وFacebook. يمكن العثور على معظم المنشورات التي تحتوي على خطاب كراهيةفي حسابات الأشخاص ذوي الآراء السياسية. لذلك، إذا كنت تريد معرفة كيفية تدريب نموذج اكتشاف خطاب الكراهية باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سأوجهك خلال مهمة اكتشاف خطاب الكراهية باستخدام التعلم الآلي باستخدام بايثون.

اكتشاف خطاب الكراهية باستخدام التعلم الآلي

لا يوجد تعريف قانوني لخطاب الكراهية لأنه لا يمكن بسهولة تصنيف آراء الأشخاص على أنها بغيضة أو مسيئة. ومع ذلك، تُعرِّف الأمم المتحدة خطاب الكراهية بأنه أي نوع من الاتصالات اللفظية أو الكتابية أو السلوكية التي يمكن أن تهاجم أو تستخدم لغة تمييزية تجاه شخص أو مجموعة من الأشخاص بناءً على هويتهم على أساس الدين أو العرق أو الجنسية أو العرق أو اللون أو النسب. أو الجنس أو أي عامل هوية آخر.

آمل أن تكون قد فهمت الآن ما هو خطاب الكراهية. تحتاج منصات وسائل التواصل الاجتماعي إلى اكتشاف خطاب الكراهية ومنع انتشاره أو حظرهفي الوقت المناسب. لذلكفي القسم أدناه، سأوجهك خلال مهمة اكتشاف الكلام الذي يحض على الكراهية باستخدام التعلم الآلي باستخدام لغة برمجة بايثون.

اكتشاف خطاب الكراهية باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة اكتشاف خطاب الكراهية من Kaggle. تم جمع مجموعة البيانات هذهفي الأصل من Twitter وتحتوي على الأعمدة التالية:

1. فهرس (index).
2. عدد (count).
3. خطاب الكراهية (hate_speech).
4. لغة بذيئة (offensive_language).
5. لا هذا ولا ذاك (neither).

6. فئة (class).

7. تغريدة (tweet).

فلنبدأ باستيراد جميع مكتبات بايثون ومجموعة البيانات اللازمة لهذه المهمة:

```
from nltk.util import pr
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import re
import nltk
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))
data = pd.read_csv("twitter.csv")
print(data.head())
```

```
   Unnamed: 0  count  hate_speech  offensive_language  neither  class  \
0           0      3            0                   0        3      2
1           1      3            0                   3        0      1
2           2      3            0                   3        0      1
3           3      3            0                   2        1      1
4           4      6            0                   6        0      1

                                               tweet
0  !!! RT @mayasolovely: As a woman you shouldn't...
1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...
```

سأضيف عمودًا جديداً إلى مجموعة البيانات هذه كعناوين تحتوي على القيم على النحو التالي:

1. خطاب الكراهية (Hate Speech).

2. لغة بذيئة (Offensive Language).

3. لا كره ومهينه (No Hate and Offensive).

```
data["labels"] = data["class"].map({0: "Hate Speech",
                                    1: "Offensive Language",
                                    2: "No Hate and
Offensive"})
print(data.head())
```

```
   Unnamed: 0  count  hate_speech  offensive_language  neither  class  \
0           0      3            0                   0        3      2
1           1      3            0                   3        0      1
2           2      3            0                   3        0      1
3           3      3            0                   2        1      1
4           4      6            0                   6        0      1

                                               tweet                labels
0  !!! RT @mayasolovely: As a woman you shouldn't...  No Hate and Offensive
1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...     Offensive Language
2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...     Offensive Language
3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...     Offensive Language
4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...     Offensive Language
```

الآن سأختار فقط أعمدة التغريدات والتسميات لبقية مهمة تدريب نموذج اكتشاف خطاب الكراهية:

```
data = data[["tweet", "labels"]]
print(data.head())
```

```
                                               tweet                labels
0  !!! RT @mayasolovely: As a woman you shouldn't...  No Hate and Offensive
1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...     Offensive Language
2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...     Offensive Language
3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...     Offensive Language
4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...     Offensive Language
```

الآن سوف أقوم بإنشاء دالة لتنظيف النصوص في عمود التغريدة (tweet column):

```
def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in
stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["tweet"] = data["tweet"].apply(clean)
```

فلنقم الآن بتقسيم مجموعة البيانات إلى مجموعات تدريب واختبار وتدريب نموذج التعلم الآلي لمهمة اكتشاف الكلام الذي يحض على الكراهية:

```
x = np.array(data["tweet"])
y = np.array(data["labels"])

cv = CountVectorizer()
```

```
X = cv.fit_transform(x) # Fit the Data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

clf = DecisionTreeClassifier()
clf.fit(X_train,y_train)
```

دعنا الآن نختبر نموذج التعلم الآلي هذا لمعرفة ما إذا كان يكتشف الكلام الذي يحض على الكراهية أم لا:

```
sample = "Let's unite and kill all the people who are
protesting against the government"
data = cv.transform([sample]).toarray()
print(clf.predict(data))
```

```
['Hate Speech']
```

الملخص

هذه هي الطريقة التي يمكنك بها تدريب نموذج التعلم الآلي على مهمة اكتشاف خطاب الكراهية باستخدام لغة برمجة بايثون. يعتبر خطاب الكراهية من القضايا الخطيرة التي نراها يوميًا على منصات التواصل الاجتماعي مثل Twitter وFacebook. يمكن العثور على معظم المنشورات التي تحتوي على خطاب كراهيةفي حسابات الأشخاص ذوي الآراء السياسية. أتمنى أن تكون قد أحببت هذه المقالة حول مهمة اكتشاف خطاب الكراهية باستخدام التعلم الآلي باستخدام بايثون.

37) التنبؤ بسعر Dogecoin مع التعلم الآلي Dogecoin Price Prediction with Machine Learning

Dogecoin هو سبب الانخفاض الأخيرفي أسعار Bitcoin. سعر Dogecoin رخيص جدًا حاليًا مقارنة بعملة Bitcoin، لكن بعض الخبراء الماليين، بمافي ذلك الرئيس التنفيذي لشركة Tesla Elon Musk، يزعمون أننا سنشهد ارتفاعًافي سعر Dogecoin قريبًا. لذا، إذاكنت تريد معرفة كيفية التنبؤ بالأسعار المستقبلية لـ Dogecoin، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة توقع أسعار Dogecoin مع التعلم الآلي باستخدام بايثون.

التنبؤ بسعر Dogecoin

يعد التنبؤ بسعر العملة المشفرة مشكلة انحدار (regression)في التعلم الآلي. تعد Bitcoin واحدة من أكثر الأمثلة نجاحًا على العملات المشفرة، لكننا شهدنا مؤخرًا انخفاضًا كبيرًافي أسعار Bitcoin بسبب Dogecoin. على عكس Bitcoin، تعتبر عملة Dogecoin رخيصة جدًافي الوقت الحالي، لكن الخبراء الماليين يتوقعون أننا قد نشهد زيادة كبيرةفي أسعار Dogecoin.

هناك العديد من مناهج التعلم الآلي التي يمكننا استخدامها لمهمة التنبؤ بسعر Dogecoin. يمكنك تدريب نموذج التعلم الآلي أو يمكنك أيضًا استخدام نموذج قوي متاح بالفعل مثل نموذج Facebook Prophet. لكنفي القسم أدناه، سأستخدم مكتبة autotsفي بايثون لمهمة التنبؤ بأسعار Dogecoin مع التعلم الآلي.

التنبؤ بسعر Dogecoin باستخدام بايثون

للتنبؤ بأسعار Dogecoin المستقبلية، تحتاج أولاً إلى الحصول على مجموعة بيانات لهذه المهمة. لذلك للحصول على مجموعة بيانات لمهمة التنبؤ بسعر Dogecoin، ما عليك سوى اتباع الخطوات المذكورة أدناه:

1. قم بزيارة Yahoo Finance.
2. ابحث عن "Dogecoin".
3. انقر فوق "البيانات التاريخية Historical Data ".
4. انقر فوق "تنزيل Download ".

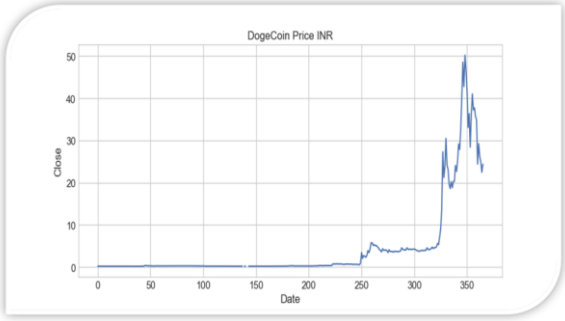
بعد الانتهاء من الخطوات المذكورة أعلاه، ستجد مجموعة بيانات بالأسعار التاريخية لـ Dogecoinفي مجلد التنزيلات الخاص بك. لنبدأ الآن بمهمة توقع أسعار Dogecoin عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from seaborn import regression
sns.set()
plt.style.use('seaborn-whitegrid')

data = pd.read_csv("Dogecoin.csv")
print(data.head())
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2020-05-24	0.193350	0.194625	0.186274	0.186783	0.186783	1.418502e+10
1	2020-05-25	0.186607	0.193194	0.185048	0.192753	0.192753	1.628989e+10
2	2020-05-26	0.192689	0.192902	0.186774	0.187698	0.187698	1.400234e+10
3	2020-05-27	0.187635	0.191591	0.187006	0.190508	0.190508	1.413078e+10
4	2020-05-28	0.190621	0.193574	0.188966	0.191035	0.191035	1.667015e+10

في مجموعة البيانات هذه، يحتوي عمود الإغلاق (close) على القيم التي نريد التنبؤ بقيمها المستقبلية، لذلك دعونا نلقي نظرة فاحصة على القيم التاريخية لأسعار إغلاق Dogecoin:

```
data.dropna()
plt.figure(figsize=(10, 4))
plt.title("DogeCoin Price INR")
plt.xlabel("Date")
plt.ylabel("Close")
plt.plot(data["Close"])
plt.show()
```



سأستخدم الآن مكتبة autots في بايثون لتدريب نموذج التعلم الآلي للتنبؤ بالأسعار المستقبلية لـ Dogecoin .إذا لم تكن قد استخدمت هذه المكتبة من قبل ، فيمكنك تثبيتها بسهولةفي نظامك باستخدام الأمرpip :

```
pip install autots
```

الآن دعنا ندرب نموذج التنبؤ بأسعار Dogecoin ونلقي نظرة على الأسعار المستقبلية لـ Dogecoin:

```
from autots import AutoTS
```

```
model = AutoTS(forecast_length=10, frequency='infer',
ensemble='simple', drop_data_older_than_periods=200)
model = model.fit(data, date_col='Date', value_col='Close',
id_col=None)

prediction = model.predict()
forecast = prediction.forecast
print("DogeCoin Price Prediction")
print(forecast)
```

```
DogeCoin Price Prediction
               Close
2021-05-25  23.625960
2021-05-26  24.655236
2021-05-27  24.642397
2021-05-28  25.270966
2021-05-29  26.182042
2021-05-30  26.204409
2021-05-31  27.254508
2021-06-01  28.709306
2021-06-02  29.425843
2021-06-03  29.497685
```

الملخص

هناك العديد من مناهج التعلم الآلي التي يمكننا استخدامها لمهمة التنبؤ بالأسعار المستقبلية لـ Dogecoin.في هذه المقالة، قدمت لك كيف يمكنك التنبؤ بالأسعار المستقبلية لـ Dogecoin باستخدام مكتبة autotsفي بايثون. آمل أن تكون قد أحببت هذه المقالة حول كيفية التنبؤ بالأسعار المستقبلية لـ Dogecoin باستخدام التعلم الآلي باستخدام بايثون.

38) تصنيف إعلانات وسائل التواصل الاجتماعي مع التعلم الآلي Social Media Ads Classification with Machine Learning

يتعلق تصنيف إعلانات الوسائط الاجتماعية (Social Media Ads Classification) بتحليل الإعلانات لتصنيف ما إذا كان جمهورك المستهدف سيشتري المنتج أم لا. إنها حالة استخدام رائعة لعلوم البياناتفي التسويق. لذلك، إذا كنت ترغبفي معرفة كيفية تحليل إعلانات الوسائط الاجتماعية لتصنيف جمهورك المستهدف، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تصنيف إعلانات الوسائط الاجتماعية مع التعلم الآلي باستخدام بايثون.

تصنيف إعلانات وسائل التواصل الاجتماعي

يعني تصنيف إعلانات الوسائط الاجتماعية تحليل إعلانات الوسائط الاجتماعية الخاصة بك للعثور على العملاء الأكثر ربحية لمنتجك والذين من المرجح أن يشتروا المنتج.في بعض الأحيان، لا يكون المنتج الذي تقدمه مناسبًا لجميع الأشخاص عندما يتعلق الأمر بالعمر والدخل. على سبيل المثال، قد يرغب الشخص الذي يتراوح عمره بين 20 و 25 عامًافي الإنفاق على أغلفة الهواتف الذكية أكثر من الشخص الذي يتراوح عمره بين 40 و 45 عامًا.

وبالمثل، يمكن للشخص ذي الدخل المرتفع أن ينفق على السلع الكمالية أكثر مما ينفقه الشخص ذو الدخل المنخفض. إذن هذه هي الطريقة التي يمكن بها لأي شركة تحديد ما إذا كان الشخص سيشتري منتجهم أم لا عن طريق تصنيف إعلانات وسائل التواصل الاجتماعي الخاصة بهم.في القسم أدناه، سوف أطلعك على تصنيف إعلانات الوسائط الاجتماعية باستخدام التعلم الآلي باستخدام بايثون.

تصنيف إعلانات وسائل التواصل الاجتماعي باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تصنيف إعلانات الوسائط الاجتماعية من Kaggle. يحتوي على بيانات حول الحملة الإعلانية لمنتج ما على وسائل التواصل الاجتماعي. يحتوي على ميزات مثل:

1. عمر الجمهور المستهدف.
2. الراتب التقديري للجمهور المستهدف.
3. وما إذا كان الجمهور المستهدف قد اشترى المنتج أم لا.

لذلك دعونا نستورد مجموعة البيانات ومكتبات بايثون الضرورية لبدء هذه المهمة:

```
import numpy
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/social.csv")
print(data.head())
```

```
   Age  EstimatedSalary  Purchased
0   19            19000          0
1   35            20000          0
2   26            43000          0
3   27            57000          0
4   19            76000          0
```

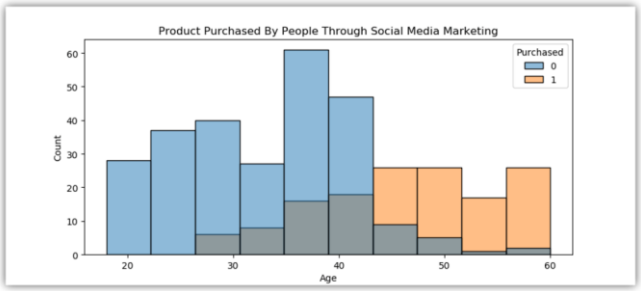
دعنا الآن نلقي نظرة على بعض الأفكار من البيانات لمعرفة ما إذا كنا بحاجة إلى إجراء أي تغييرات على مجموعة البيانات:

```
print(data.describe())
print(data.isnull().sum())
```

```
              Age  EstimatedSalary   Purchased
count  400.000000       400.000000  400.000000
mean    37.655000     69742.500000    0.357500
std     10.482877     34096.960282    0.479864
min     18.000000     15000.000000    0.000000
25%     29.750000     43000.000000    0.000000
50%     37.000000     70000.000000    0.000000
75%     46.000000     88000.000000    1.000000
max     60.000000    150000.000000    1.000000
Age               0
EstimatedSalary   0
Purchased         0
dtype: int64
```

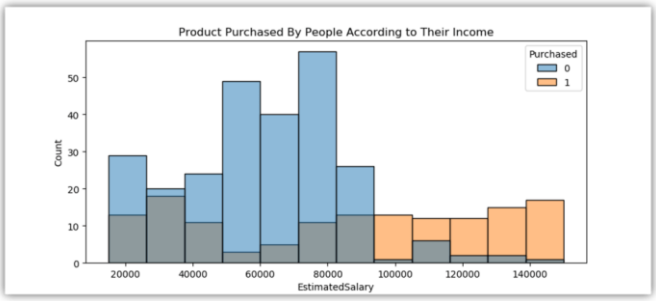
دعنا الآن نستكشف بعض الأنماط المهمةفي مجموعة البيانات. أول ما أريد استكشافه هو أعمار الأشخاص الذين استجابوا لإعلانات وسائل التواصل الاجتماعي واشتروا المنتج:

```
plt.figure(figsize=(15, 10))
plt.title("Product Purchased By People Through Social Media Marketing")
sns.histplot(x="Age", hue="Purchased", data=data)
plt.show()
```



يوضح الرسم أعلاه أن الأشخاص الذين تزيد أعمارهم عن 45 عامًا من بين الجمهور المستهدف يهتمون أكثر بشراء المنتج. دعنا الآن نلقي نظرة على فئة الدخل للأشخاص الذين استجابوا لإعلانات الشبكات الاجتماعية واشتروا المنتج:

```
plt.title("Product Purchased By People According to Their
Income")
sns.histplot(x="EstimatedSalary", hue="Purchased", data=data)
plt.show()
```



يوضح الرسم أعلاه أن الأشخاص الذين يزيد دخلهم الشهري عن 90.000 من بين الجمهور المستهدف يهتمون أكثر بشراء المنتج.

تدريب نموذج تصنيف إعلانات وسائل التواصل الاجتماعية

دعنا الآن ندرب نموذجًا لتصنيف إعلانات الوسائط الاجتماعية. أولاً، سأقوم بتعيين العمود "Purchased"في مجموعة البيانات كمتغير مستهدف والعمودين الآخرين كميزات نحتاجها لتدريب نموذج:

```
x = np.array(data[["Age", "EstimatedSalary"]])
y = np.array(data[["Purchased"]])
```

لنقم الآن بتقسيم البيانات وتدريب نموذج تصنيف إعلانات الشبكات الاجتماعية باستخدام مصنف شجرة القرار (decision tree classifier):

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y,

test_size=0.10,
```

```
random_state=42)
model = DecisionTreeClassifier()
model.fit(xtrain, ytrain)
predictions = model.predict(xtest)
```

أخيرًا، دعنا نلقي نظرة على تقرير تصنيف (classification report) النموذج:

```
print(classification_report(ytest, predictions))
```

	precision	recall	f1-score	support
0	0.88	0.85	0.87	27
1	0.71	0.77	0.74	13
accuracy			0.82	40
macro avg	0.80	0.81	0.80	40
weighted avg	0.83	0.82	0.83	40

الملخص

إذن هذه هي الطريقة التي يمكنك بها تحليل إعلانات وسائل التواصل الاجتماعي وتصنيفها حول حملة التسويق لمنتج ما. يعني تصنيف إعلانات الوسائط الاجتماعية تحليل إعلانات وسائل التواصل الاجتماعي الخاصة بك للعثور على العملاء الأكثر ربحية لمنتجك والذين من المرجح أن يشتروا المنتج. آمل أن تكون قد أحببت هذه المقالة حول تصنيف إعلانات وسائل التواصل الاجتماعي باستخدام التعلم الآلي باستخدام بايثون.

39) اكتشاف الأخبار الكاذبة باستخدام التعلم الآلي Fake News Detection with Machine Learning

الأخبار الكاذبة (Fake news) هي واحدة من أكبر المشكلات التي تواجه وسائل التواصل الاجتماعي عبر الإنترنت وحتى بعض المواقع الإخبارية.في معظم الأوقات، نرى الكثير من الأخبار الكاذبة حول السياسة. لذا فإن استخدام التعلم الآلي لاكتشاف الأخبار الكاذبة يعد مهمة صعبة للغاية. إذا كنت تريد معرفة كيفية اكتشاف الأخبار الكاذبة باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة كشف الأخبار الوهمية باستخدام التعلم الآلي باستخدام بايثون.

اكتشاف الأخبار الكاذبة

الأخبار الكاذبة هي واحدة من أكبر المشاكل لأنها تؤدي إلى الكثير من المعلومات الخاطئةفي منطقة معينة.في معظم الأحيان، قد يؤدي نشر أخبار كاذبة حول المعتقدات السياسية والدينية لمجتمع ما إلى أعمال شغب وأعمال عنف كما رأيتفي البلد الذي تعيش فيه. لذلك، لاكتشاف الأخبار الكاذبة، يمكننا العثور على علاقات بين عناوين الأخبار الكاذبة حتى نتمكن من تدريب نموذج للتعلم الآلي يمكنه إخبارنا ما إذا كانت معلومة معينة كاذبة أم حقيقية بمجرد ملاحظة العنوان الرئيسيفي الأخبار. لذلكفي القسم أدناه، سأقدم لكم مشروع التعلم الآلي عن اكتشاف الأخبار الكاذبة باستخدام لغة برمجة بايثون.

اكتشاف الأخبار الكاذبة باستخدام بايثون

تحتوي مجموعة البيانات التي أستخدمها هنا لمهمة الكشف عن الأخبار الكاذبة على بيانات حول عنوان الأخبار ومحتوى الأخبار وعمود يُعرف باسم التسمية (label) يوضح ما إذا كانت الأخبار كاذبة أم حقيقية. لذلك يمكننا استخدام مجموعة البيانات هذه للعثور على العلاقات بين عناوين الأخبار الكاذبة والحقيقية لفهم نوع العناوين الرئيسيةفي معظم الأخبار الكاذبة. لذلك دعونا نستورد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لهذه المهمة:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB

data = pd.read_csv("news.csv")
print(data.head())
```

```
   Unnamed: 0  ... label
0        8476  ...  FAKE
1       10294  ...  FAKE
2        3608  ...  REAL
3       10142  ...  FAKE
4         875  ...  REAL
```

مجموعة البيانات هذه كبيرة جدًا ولحسن الحظ لا تزال لا تحتوي على قيم مفقودة، لذا دون إضاعة أي وقت، دعنا نستخدم عمود العنوان (title column) كميزة نحتاجها لتدريب نموذج التعلم الآلي وعمود التسمية (label column) على أنهما القيم التي نريد توقعها:

```
x = np.array(data["title"])
y = np.array(data["label"])

cv = CountVectorizer()
x = cv.fit_transform(x)
```

دعنا الآن نفصل مجموعة البيانات إلى مجموعات تدريب واختبار، وبعد ذلك سأستخدم خوارزمية Multinomial Naive Bayes لتدريب نموذج اكتشاف الأخبار الكاذبة:

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.2, random_state=42)
model = MultinomialNB()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

```
0.8074191002367798
```

الآن دعونا نختبر هذا النموذج. لاختبار نموذجنا المدرّب، سأقوم أولاً بتدوين عنوان أي خبر موجودفي أخبار google لمعرفة ما إذا كان نموذجنا يتوقع أن الأخبار حقيقية أم لا:

```
news_headline = "CA Exams 2021: Supreme Court asks ICAI to
extend opt-out option for July exams, final order tomorrow"
data = cv.transform([news_headline]).toarray()
print(model.predict(data))
```

```
['REAL']
```

سأقوم الآن بكتابة عنوان أخبار كاذف عشوائي لمعرفة ما إذا كان النموذج يتنبأ بأن الأخبار كاذبة أم لا:

```
news_headline = "Cow dung can cure Corona Virus"
data = cv.transform([news_headline]).toarray()
print(model.predict(data))
```

```
['FAKE']
```

الملخص

هذه هي الطريقة التي يمكننا بها تدريب نموذج التعلم الآلي لمهمة اكتشاف الكاذبة باستخدام لغة برمجة بايثون. الأخبار الكاذبة هي واحدة من أكبر المشاكل لأنها تؤدي إلى الكثير من المعلومات الخاطئةفي منطقة معينة. أتمنى أن تكون قد أحببت هذه المقالة حول مهمة اكتشاف الأخبار الكاذبة باستخدام التعلم الآلي باستخدام بايثون.

40) التنبؤ بالمبيعات باستخدام التعلم الآلي Sales Prediction with Machine Learning

يعد التنبؤ بالمبيعات (Forecasting sales) مشكلة صعبة لكل نوع من أنواع الأعمال، ولكنه يساعدفي تحديد المكان الذي يجب أن تنفق فيه الشركة أكثر على الإعلان وأين يجب أن تقلل الإنفاق.في هذه المقالة، سوف أطلعك على مهمة التنبؤ بالمبيعات باستخدام التعلم الآلي باستخدام بايثون.

ما هو التنبؤ بالمبيعات؟

يعني التنبؤ بالمبيعات التنبؤ بكمية المنتج التي سيشتريها الأشخاص بناءً على عوامل مثل المبلغ الذي تنفقه للإعلان عن منتجك، أو شريحة الأشخاص الذين تعلن عنهم، أو النظام الأساسي الذي تعلن عنه حول منتجك.

عادةً ما تعتمد مبيعات المنتج بشكل أساسي على المبلغ الذي تنفقه على الإعلان عنه، فكلما زاد عدد الأشخاص الذين يصل منتجك، زادت المبيعات طالما أن جودة منتجك جيدة. الآنفي القسم أدناه، سوف آخذك خلال مشروع التعلم الآلي حول التنبؤ بالمبيعات باستخدام بايثون.

التنبؤ بالمبيعات باستخدام بايثون

آمل أن تفهم الآن ما هو التنبؤ بالمبيعات. عادةً ما يحتاج النشاط التجاري القائم على المنتج والخدمات دائمًا إلى عالم البيانات الخاص به للتنبؤ بمبيعاته المستقبلية مع كل خطوة يتخذها للتلاعب بتكلفة الإعلان عن منتجهم. فلنبدأ مهمة التنبؤ بالمبيعات باستخدام التعلم الآلي باستخدام بايثون. سأبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/advertising.csv")
print(data.head())
```

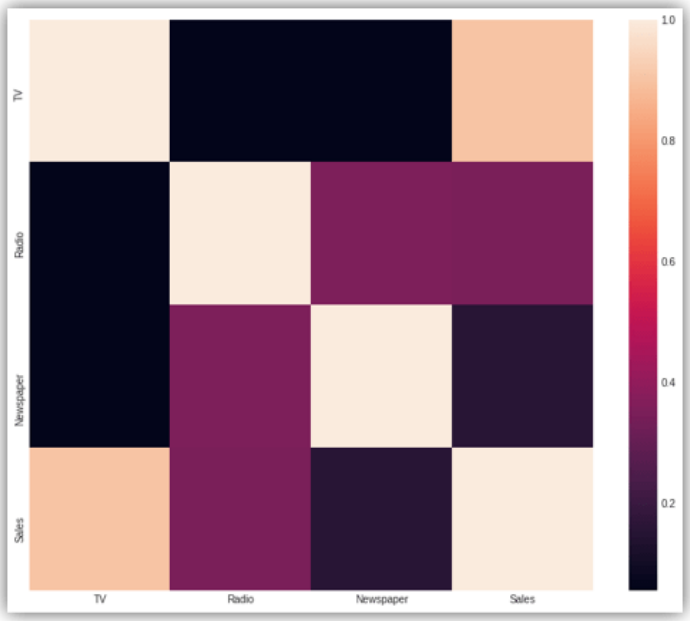
```
      TV  Radio  Newspaper  Sales
0  230.1   37.8       69.2   22.1
1   44.5   39.3       45.1   10.4
2   17.2   45.9       69.3   12.0
3  151.5   41.3       58.5   16.5
4  180.8   10.8       58.4   17.9
```

```
print(data.isnull().sum())
```

```
TV          0
Radio       0
Newspaper   0
Sales       0
dtype: int64
```

لذلك لا تحتوي مجموعة البيانات هذه على أي قيم فارغة. دعنا الآن نلقي نظرة على الارتباط بين الميزات قبل أن نبدأفي تدريب نموذج التعلم الآلي للتنبؤ بالمبيعات المستقبلية:

```
plt.style.use('seaborn-whitegrid')
plt.figure(figsize=(12, 10))
sns.heatmap(data.corr())
plt.show()
```



دعنا الآن نجهز البيانات لتلائم نموذج التعلم الآلي ثم سأستخدم خوارزمية الانحدار الخطي (linear regression) لتدريب نموذج التنبؤ بالمبيعات باستخدام بايثون:

```
x = np.array(data.drop(["Sales"], 1))
y = np.array(data["Sales"])
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.2, random_state=42)
model = LinearRegression()
model.fit(xtrain, ytrain)
ypred = model.predict(xtest)

data = pd.DataFrame(data={"Predicted Sales": ypred.flatten()})
print(data)
```

	Predicted Sales
0	17.034772
1	20.409740
2	23.723989
3	9.272785
4	21.682719
5	12.569402
6	21.081195
7	8.690350
8	17.237013
9	16.666575
10	8.923965
11	8.481734
12	18.207512
13	8.067507
14	12.645510
15	14.931628
16	8.128146
17	17.898766
18	11.008806
19	20.478328
20	20.806318
21	12.598833
22	10.905183
23	22.388548
24	9.417961
25	7.925067
26	20.839085
27	13.815209
28	10.770809
29	7.926825
30	15.959474
31	10.634909
32	20.802920
33	10.434342
34	21.578475
35	21.183645
36	12.128218
37	22.809533
38	12.609928
39	6.464413

الملخص

هذه هي الطريقة التي يمكننا بها التنبؤ بالمبيعات المستقبلية لمنتج ما باستخدام التعلم الآلي. يعد التنبؤ بالمبيعات مشكلة صعبة لكل نوع من أنواع الأعمال ولكن استخدام التعلم الآلي يجعل الأمر يبدو سهلاً. آمل أن تكون قد أحببت هذه المقالة حول مهمة التنبؤ بالمبيعات باستخدام التعلم الآلي باستخدام بايثون.

41) تصنيف أسعار الأجهزة المحمولة مع التعلم الآلي Mobile Price Classification with Machine Learning

سعر المنتج هو أهم سمة لتسويق هذا المنتج. أحد تلك المنتجات التي يكون السعر فيها مهمًا كثيرًا هو الهاتف الذكي لأنه يأتي مع الكثير من الميزات بحيث تفكر الشركة كثيرًافي كيفية تسعير هذا الهاتف المحمول الذي يمكن أن يبرر الميزات ويغطي أيضًا تكاليف التسويق والتصنيع للهاتف المحمول.في هذه المقالة، سوف أطلعك على مهمة تصنيف أسعار الأجهزة المحمولة باستخدام التعلم الآلي باستخدام بايثون.

تصنيف أسعار الأجهزة المحمولة مع التعلم الآلي

الهواتف المحمولة هي الأجهزة الإلكترونية الأكثر مبيعًا حيث يواصل الأشخاص تحديث هواتفهم المحمولة كلما وجدوا ميزات جديدةفي جهاز جديد. يتم بيع الآلاف من الهواتف المحمولة يوميًا، وفي مثل هذه الحالة يكون من الصعب جدًا على الشخص الذي يخطط لإنشاء أعمال الهاتف المحمول الخاصة به أن يقرر السعر الذي يجب أن يكون عليه الهاتف المحمول.

في القسم أدناه، سأقدم لك مشروع التعلم الآلي على نموذج تصنيف أسعار الأجهزة المحمولة حيث سأقوم بتدريب نموذج لتصنيف النطاق السعري للهواتف المحمولة باستخدام بايثون. تعتمد هذه المهمة على حل دراسة الحالة المذكورة أدناه.

" يريد السيد أمان أن يبدأ شركة الهواتف المحمولة الخاصة به ويريد أن يخوض معركة شاقة مع العلامات التجارية الكبرى للهواتف الذكية مثل Samsung وApple. لكنه لا يعرف كيفية تقدير سعر الهاتف المحمول الذي يمكن أن يغطي كلاً من تكاليف التسويق والتصنيع. لذافي هذه المهمة، لا يتعين عليك توقع الأسعار الفعلية للهواتف المحمولة ولكن عليك توقع النطاق السعري للهواتف المحمولة. "

تصنيف أسعار الأجهزة المحمولة باستخدام لغة بايثون

لذلك، نظرًا لأن مهمتنا هي تصنيف النطاق السعري للهواتف المحمولة وليس التنبؤ بالأسعار الفعلية، لذلك سأقوم هنا بتدريب نموذج تصنيف لتصنيف النطاق السعري للهواتف المحمولة على النحو التالي:

- 0 (تكلفة منخفضة).
- 1 (تكلفة متوسطة).
- 2 (تكلفة عالية).

- 4 (تكلفة عالية جدا).

سأبدأ مهمة تصنيف أسعار الأجهزة المحمولة باستخدام التعلم الآلي عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

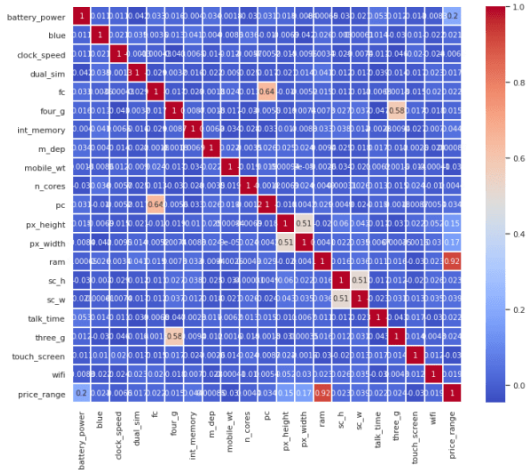
مجموعة البيانات

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
sns.set()

data = pd.read_csv("mobile_prices.csv")
print(data.head())
```

```
   battery_power  blue  clock_speed  ...  touch_screen  wifi  price_range
0            842     0          2.2  ...             0     1            1
1           1021     1          0.5  ...             1     0            2
2            563     1          0.5  ...             1     0            2
3            615     1          2.5  ...             0     0            2
4           1821     1          1.2  ...             1     0            1

[5 rows x 21 columns]
```

لذا، تحتوي مجموعة البيانات على 21 عمودًا ولحسن الحظ لا تحتوي مجموعة البيانات هذه على قيم مفقودة، لذا يمكننا البدء فقط من خلال تدريب النموذج، ولكن قبل ذلك، دعنا نلقي نظرة على الارتباط بين الميزاتفي مجموعة البيانات:

```
plt.figure(figsize=(12, 10))
sns.heatmap(data.corr(), annot=True, cmap="coolwarm",
linecolor='white', linewidths=1)
```



تحضير البيانات

لا تحتوي مجموعة البيانات هذه على ميزات فئوية، لذلك يمكننا فقط استخدام هذه البيانات دون أي تحويل لأن جميع ميزات مجموعة البيانات رقمية. ولكن لتدريب نموذج، من المهم جدًا توحيد البيانات أو تطبيعها وتقسيمها إلى مجموعات تدريب واختبار.

لذلك دعونا نوحد مجموعة البيانات ونقسم البيانات إلى 80٪ تدريب و20٪ اختبار:

```
x = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
x = StandardScaler().fit_transform(x)
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.20, random_state=0)
```

نموذج تصنيف سعر الهاتف المحمول

دعنا الآن ندرب نموذج تصنيف أسعار الجوّال باستخدام بايثون. نظرًا لأن هذه مشكلة تصنيف، فسوف أستخدم خوارزمية الانحدار اللوجستي Logistic Regression المقدمة من Scikit-Learn:

```
from sklearn.linear_model import LogisticRegression
lreg = LogisticRegression()
lreg.fit(x_train, y_train)
y_pred = lreg.predict(x_test)
```

الآن دعنا نلقي نظرة على دقة النموذج:

```
accuracy = accuracy_score(y_test, y_pred) * 100
print("Accuracy of the Logistic Regression Model: ",accuracy)
```

```
Accuracy of the Logistic Regression Model:  95.5
```

لذا فإن النموذج يعطي دقة تبلغ حوالي 95.5٪ وهو أمر رائع. دعنا الآن نلقي نظرة على التنبؤات التي قدمها النموذج:

```
[3 0 2 2 3 0 0 3 3 1 1 3 0 2 3 0 3 2 2 1 0 0 3 1 2 2 3 1 3 1 1 0 2 0 2 3 0
 0 3 3 3 1 3 3 1 3 0 1 3 1 1 3 0 3 0 2 2 2 0 3 3 1 3 2 1 2 3 2 2 2 3 2 1 0
 1 3 2 2 1 2 3 3 3 0 0 0 2 1 2 3 1 2 2 1 0 3 3 3 0 3 1 1 3 1 3 2 2 3 2 3 3
 0 0 1 3 3 0 0 1 0 0 3 2 2 1 2 1 1 1 0 2 1 3 3 3 3 3 3 2 0 1 1 2 1 3 0 3 0 0
 2 0 1 1 1 1 3 0 0 3 1 3 2 1 3 1 2 3 3 3 2 1 0 3 1 2 3 3 0 2 2 3 1 2 1 0 1 2
 2 2 0 3 3 1 1 0 2 3 0 1 2 2 0 3 3 3 1 2 3 3 3 0 0 0 2 3 3 0 0 1 3 2 3 3 3
 0 0 2 3 3 1 0 2 0 0 0 3 2 1 2 2 1 1 0 2 3 3 0 0 1 3 3 1 3 0 3 1 1 0 2 3 3
 2 0 0 1 2 3 2 2 3 2 1 0 3 3 2 1 3 2 2 2 1 0 2 2 1 0 0 2 2 2 3 0 1 3 0 2 2
 3 0 2 0 1 1 3 0 0 2 3 1 2 0 2 0 3 0 3 3 2 3 1 2 2 1 1 1 0 1 0 3 1 0 3 0 0
 1 3 0 3 1 1 0 1 3 0 2 1 1 2 1 1 0 2 0 0 3 1 2 3 3 2 2 0 3 2 2 1 3 2 3 3 3 0
 2 0 3 0 1 1 2 3 1 3 1 2 0 1 2 3 0 0 1 3 0 3 0 2 2 1 1 0 2 0]
```

يوضح الناتج أعلاه النطاق السعري المصنف بواسطة النموذج. دعونا نلقي نظرة على عدد الهواتف المحمولة المصنفة لكل نطاق سعري:

```
(unique, counts) = np.unique(y_pred, return_counts=True)
price_range = np.asarray((unique, counts)).T
print(price_range)
```

أتمنى أن تكون قد أحببت هذه المقالة حول تصنيف أسعار الأجهزة المحمولة باستخدام التعلم الآلي.

43) التنبؤ بسعر صرف العملات مع التعلم الآلي Currency Exchange Rate Prediction with Machine Learning

صرف العملات (Currency exchange) هو واحد من أكبر الأسواق المالية. حاليا، دولار واحد من دولارات الولايات المتحدة يعادل 73.02 روبية هندية. تؤثر العديد من العوامل على أسعار الصرف مثل العوامل الاقتصادية والسياسية وحتى النفسية. يعد التنبؤ بسعر صرف العملات (Currency Exchange Rate Prediction) مشكلة صعبة، لذافي هذه المقالة، سوف أطلعك على مهمة التنبؤ بسعر صرف العملات باستخدام التعلم الآلي باستخدام بايثون.

التنبؤ بسعر صرف العملات

التنبؤ بأسعار صرف العملات هو مشكلة انحدار (regression)في التعلم الآلي. هناك تغيراتفي أسعار الصرف كل يوم تؤثر على دخل الفرد أو الشركة ويمكن أن تؤثر حتى على اقتصاد بلد ما. وبالتالي، فإن التنبؤ بأسعار صرف العملات يمكن أن يساعد الفرد وكذلك البلدفي نواح كثيرة.

هناك العديد من خوارزميات التعلم الآلي التي يمكننا استخدامها للتنبؤ بأسعار صرف العملاتفي المستقبل. يمكنك أيضًا استخدام الشبكات العصبية الاصطناعية (artificial neural networks) لهذه المهمة.في القسم أدناه، سوف آخذك خلال مهمة التنبؤ بسعر صرف العملات مع التعلم الآلي باستخدام بايثون.

التنبؤ بسعر صرف العملات باستخدام بايثون

للتنبؤ بسعر صرف العملات من خلال التعلم الآلي، نحتاج أولاً إلى الحصول على البيانات الأكثر ملاءمة لهذه المهمة. للحصول على مجموعة بيانات لهذه المهمة، ما عليك سوى اتباع الخطوات المذكورة أدناه:

1. قم بزيارة Yahoo Finance.
2. ابحث عن "USD / INR (INR = x)".
3. انقر فوق "البيانات التاريخية Historical Data "
4. انقر فوق "تنزيل Download "

باتباع الخطوات المذكورة أعلاه، ستتمكن من تنزيل البيانات التاريخية لأسعار صرف العملات بالروبية الهندية. بعد النقر فوق تنزيل، ستتلقى ملف CSVفي مجلد التنزيلات الخاص بك.

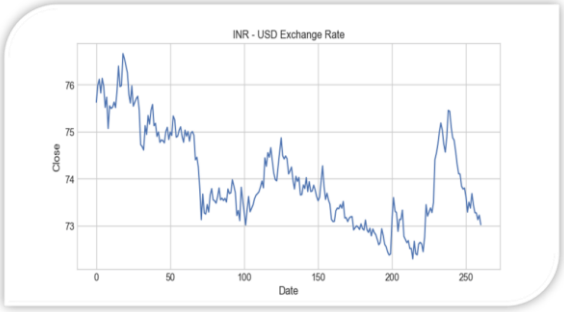
دعنا الآن نستورد مكتبات بايثون الضرورية التي نحتاجها لهذه المهمة ونقرأ مجموعة البيانات:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from seaborn import regression
sns.set()
plt.style.use('seaborn-whitegrid')

data = pd.read_csv("INR.csv")
print(data.head())
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2020-05-22	75.625000	76.209503	75.610001	75.625000	75.625000	0
1	2020-05-25	75.985001	76.129501	75.757500	75.985001	75.985001	0
2	2020-05-26	75.873596	76.110001	75.404999	76.110001	76.110001	0
3	2020-05-27	75.489502	76.000000	75.381302	75.820000	75.820000	0
4	2020-05-28	75.885696	76.129997	75.634499	76.129997	76.129997	0

في مجموعة البيانات هذه، القيم الموجودةفي عمود الاغلاق "Close" هي القيم المستهدفة التي نحتاج إلى توقعها. لذلك دعونا نلقي نظرة فاحصة على هذه القيم:

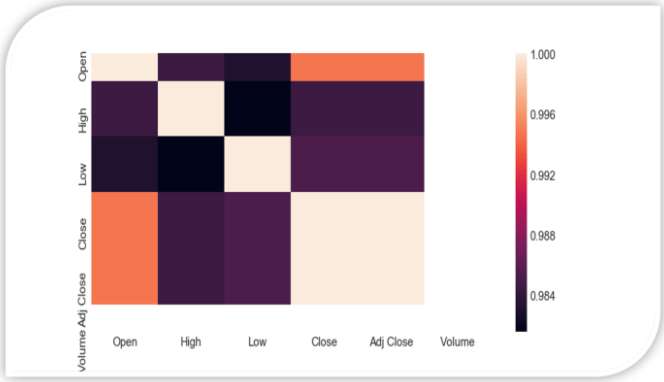
```
plt.figure(figsize=(10, 4))
plt.title("INR - USD Exchange Rate")
plt.xlabel("Date")
plt.ylabel("Close")
plt.plot(data["Close"])
plt.show()
```



دعنا الآن نلقي نظرة على الارتباط بين الميزات قبل تدريب نموذج التنبؤ بسعر صرف العملات:

```
print(data.corr())
sns.heatmap(data.corr())
plt.show()
```

	Open	High	Low	Close	Adj Close	Volume
Open	1.000000	0.984518	0.983143	0.994720	0.994720	NaN
High	0.984518	1.000000	0.981582	0.984599	0.984599	NaN
Low	0.983143	0.981582	1.000000	0.985281	0.985281	NaN
Close	0.994720	0.984599	0.985281	1.000000	1.000000	NaN
Adj Close	0.994720	0.984599	0.985281	1.000000	1.000000	NaN
Volume	NaN	NaN	NaN	NaN	NaN	NaN



الآن الخطوة التالية هي إعداد مجموعة البيانات عن طريق تخزين الميزات الأكثر صلةفي المتغير x وتخزين العمود الهدفي المتغير y:

```
x = data[["Open", "High", "Low"]]
y = data["Close"]
x = x.to_numpy()
y = y.to_numpy()
y = y.reshape(-1, 1)
```

الآن، دعنا نقسم مجموعة البيانات ونقوم بتدريب نموذج توقع تبادل العملات باستخدام نموذج انحدار شجرة القرار (Decision Tree Regression) باستخدام لغة بايثون:

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.2, random_state=42)

from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor()
model.fit(xtrain, ytrain)
ypred = model.predict(xtest)
```

دعنا الآن نلقي نظرة على القيم المتوقعة لأسعار صرف العملات بالروبية الهندية للأيام الخمسة القادمة:

```
data = pd.DataFrame(data={"Predicted Rate": ypred.flatten()})
print(data.head())
```

```
   Predicted Rate
0       74.820000
1       74.019997
2       73.089203
3       73.374802
4       73.133400
```

الملخص

التنبؤ بأسعار صرف العملات هو مشكلة الانحدارفي التعلم الآلي.في هذه المقالة، استخدمت خوارزمية انحدار شجرة القرار للتنبؤ بأسعار صرف العملات. يمكنك استخدام خوارزميات الانحدار الأخرى وحتى الشبكات العصبية الاصطناعية لهذه المهمة. آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بسعر صرف العملات باستخدام التعلم الآلي باستخدام بايثون.

44) التنبؤ بالربح مع التعلم الآلي Profit Prediction with Machine Learning

يجب أن تحدد الشركة دائمًا هدفًا يمكن تحقيقه، وإلا فلن يتمكن الموظفون من العمل بأفضل إمكاناتهم إذا وجدوا أن الهدف الذي حددته الشركة غير قابل للتحقيق. مهمة التنبؤ بالربح (profit prediction) لفترة معينة هي نفسها تحديد الأهداف. إذا كنت تعرف مقدار الربح الذي يمكنك تحقيقه من مقدار البحث والتطوير والتسويق الذي تقوم به، فيمكن أن تحقق الشركة أكثر من الربح المتوقع بشرط أن تكون القيمة المتوقعة قابلة للتحقيق. لذلكفي هذه المقالة، سوف آخذك خلال مهمة التنبؤ بالربح مع التعلم الآلي باستخدام بايثون.

التنبؤ بالربح مع التعلم الآلي

يعتمد الربح الذي تحققه الشركة لفترة معينة على عدة عوامل مثل مقدار الوقت والمال الذي تنفقه الشركة على البحث والتطوير والتسويق وغير ذلك الكثير. لذا من أجل التنبؤ بأرباح الشركة لفترة معينة، نحتاج إلى تدريب نموذج التعلم الآلي باستخدام مجموعة بيانات تحتوي على بيانات تاريخية حول الأرباح التي حققتها الشركة.

مهمة التنبؤ بالربح هي عملية مهمة لكل عمل لتحديد هدف قابل للتحقيق. على سبيل المثال، إذا أنفق النشاط التجاري 500 دولار أمريكي على التسويق، فلا يمكنه توقع ربح قدره 20000 دولار أمريكي. وبالمثل، هناك العديد من العوامل الأخرى التي يعتمد عليها ربح العمل. لذلك يجب على الشركة تحديد هدف يمكن تحقيقه.في القسم أدناه، سوف أطلعك على مهمة التنبؤ بالربح من خلال التعلم الآلي باستخدام بايثون.

التنبؤ بالربح باستخدام بايثون

تتضمن مجموعة البيانات التي أستخدمها لمهمة التنبؤ بالربح بيانات حول إنفاق البحث والتطوير، وتكلفة الإدارة، والإنفاق على التسويق، وحالة التشغيل، والأرباح التاريخية الناتجة عن 50 شركة ناشئة. فلنبدأ بمهمة توقع الربح عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("Startups.csv")
print(data.head())
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

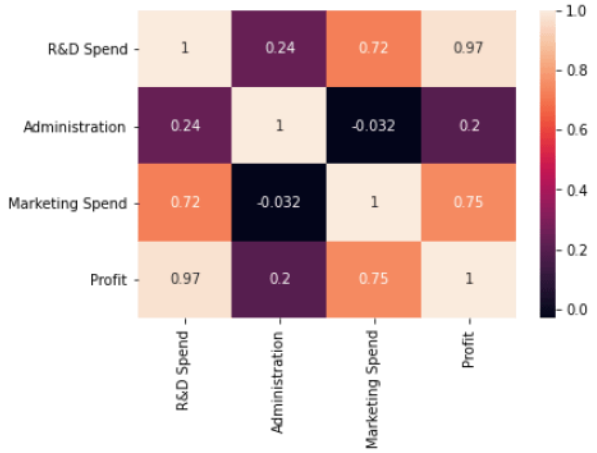
لا تحتوي هذه البيانات على أي قيم مفقودة، لذا دون إضاعة أي وقت، فلنبدأ بإلقاء نظرة على ملخص الإحصائيات لهذه البيانات:

```
print(data.describe())
```

	R&D Spend	Administration	Marketing Spend	Profit
count	50.000000	50.000000	50.000000	50.000000
mean	73721.615600	121344.639600	211025.097800	112012.639200
std	45902.256482	28017.802755	122290.310726	40306.180338
min	0.000000	51283.140000	0.000000	14681.400000
25%	39936.370000	103730.875000	129300.132500	90138.902500
50%	73051.080000	122699.795000	212716.240000	107978.190000
75%	101602.800000	144842.180000	299469.085000	139765.977500
max	165349.200000	182645.560000	471784.100000	192261.830000

الآن دعنا نلقي نظرة على الارتباط (correlation) بين الميزات:

```
sns.heatmap(data.corr(), annot=True)
plt.show()
```



نظرًا لأن هذه المهمة تعتمد على مشكلة الانحدار (regression)، فسوف أستخدم خوارزمية الانحدار الخطي (Linear regression) لتدريب نموذج التنبؤ بالربح. لذلك دعونا نجهز البيانات حتى نتمكن من ملاءمتها مع النموذج:

```
x = data[["R&D Spend", "Administration", "Marketing Spend"]]
y = data["Profit"]

x = x.to_numpy()
```

```
y = y.to_numpy()
y = y.reshape(-1, 1)

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.2, random_state=42)
```

الآن دعنا ندرب نموذج الانحدار الخطي على هذه البيانات ونلقي نظرة على القيم المتوقعة:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(xtrain, ytrain)
ypred = model.predict(xtest)
data = pd.DataFrame(data={"Predicted Profit":
ypred.flatten()})
print(data.head())
```

```
   Predicted Profit
0     126703.027165
1      84894.750816
2      98893.418160
3      46501.708150
4     129128.397344
```

الملخص

إذن هذه هي الطريقة التي يمكننا بها التنبؤ بأرباح الشركة لفترة معينة باستخدام خوارزميات التعلم الآلي. يمكن أن تساعد مثل هذه المهام الشركة على تحديد هدف يمكن تحقيقه. آمل أن تكون قد أحببت هذه المقالة حول مهمة التنبؤ بالربح مع التعلم الآلي باستخدام بايثون.

45) تحليل جودة المياه باستخدام التعلم الآلي Water Quality Analysis using Machine Learning

يُعد الحصول على مياه الشرب المأمونة أحد الاحتياجات الأساسية لجميع البشر. من وجهة نظر قانونية، يعتبر الحصول على مياه الشرب أحد حقوق الإنسان الأساسية. تؤثر العديد من العوامل على جودة المياه (Water Quality)، كما أنها أحد مجالات البحث الرئيسيةفي التعلم الآلي. لذلك إذا كنت تريد معرفة كيفية إجراء تحليل جودة المياه (water quality analysis) باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على تحليل جودة المياه باستخدام التعلم الآلي باستخدام بايثون.

تحليل جودة المياه

يُعد تحليل جودة المياه أحد المجالات الرئيسية للبحثفي التعلم الآلي. يُعرف أيضًا باسم تحليل قابلية المياه للشرب (water potability analysis) لأن مهمتنا هنا هي فهم جميع العوامل التي تؤثر على قابلية المياه للشرب وتدريب نموذج التعلم الآلي الذي يمكنه تصنيف ما إذا كانت عينة مياه معينة آمنة أو غير صالحة للاستهلاك.

بالنسبة لمهمة تحليل جودة المياه، سأستخدم مجموعة بيانات Kaggle التي تحتوي على بيانات حول جميع العوامل الرئيسية التي تؤثر على قابلية المياه للشرب. جميع العوامل التي تؤثر على جودة المياه مهمة للغاية، لذلك نحتاج إلى استكشاف كل ميزة من ميزات مجموعة البيانات هذه بإيجاز قبل تدريب نموذج التعلم الآلي للتنبؤ بما إذا كانت عينة المياه آمنة أو غير مناسبة للاستهلاك. يمكنك تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل جودة المياه من هنا.

تحليل جودة المياه باستخدام لغة بايثون

سأبدأ مهمة تحليل جودة المياه عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import numpy as np

data = pd.read_csv("water_potability.csv")
data.head()
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

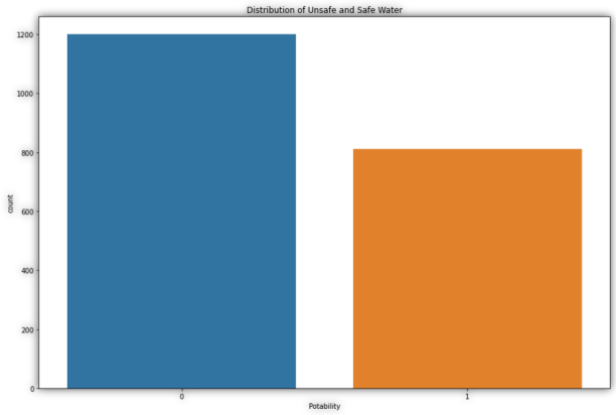
يمكنني رؤية القيم الخاليةفي المعاينة الأولى لمجموعة البيانات هذه نفسها، لذلك قبل المضي قدمًا، دعنا نزيل جميع الصفوف التي تحتوي على قيم فارغة:

```
data = data.dropna()
data.isnull().sum()
```

```
ph               0
Hardness         0
Solids           0
Chloramines      0
Sulfate          0
Conductivity     0
Organic_carbon   0
Trihalomethanes  0
Turbidity        0
Potability       0
dtype: int64
```

عمود القابلية للشرب (Potability column) لمجموعة البيانات هذه هو العمود الذي نحتاج إلى توقعه لأنه يحتوي على القيمتين 0 و 1 التي تشير إلى ما إذا كانت المياه صالحة للشرب (1) أو غير صالحة (0) للشرب. لذلك دعونا نرى توزيع 0 و 1في عمود "Potability ":

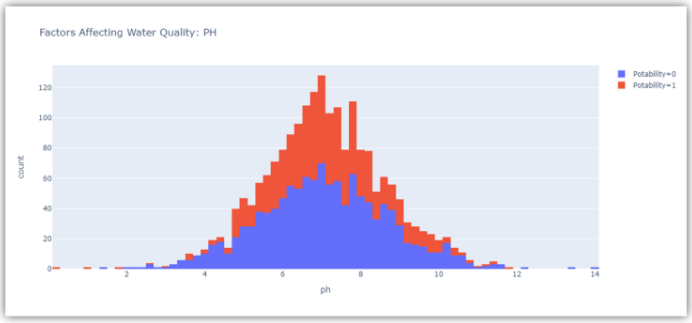
```
plt.figure(figsize=(15, 10))
sns.countplot(data.Potability)
plt.title("Distribution of Unsafe and Safe Water")
plt.show()
```



لذلك هذا شيء يجب أن تلاحظه أن مجموعة البيانات هذه غير متوازنة (not balanced) لأن عينات الأصفار أكثر من 1 ثانية.

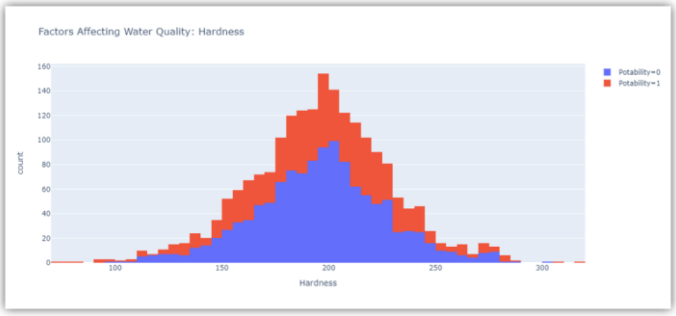
كما ذكرنا أعلاه، لا توجد عوامل لا يمكننا تجاهلها والتي تؤثر على جودة المياه، لذلك دعونا نستكشف جميع الأعمدة واحدة تلو الأخرى. لنبدأ بإلقاء نظرة على عمود الاس الهيدروجيني (ph column):

```
import plotly.express as px
data = data
figure = px.histogram(data, x = "ph",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
PH")
figure.show()
```



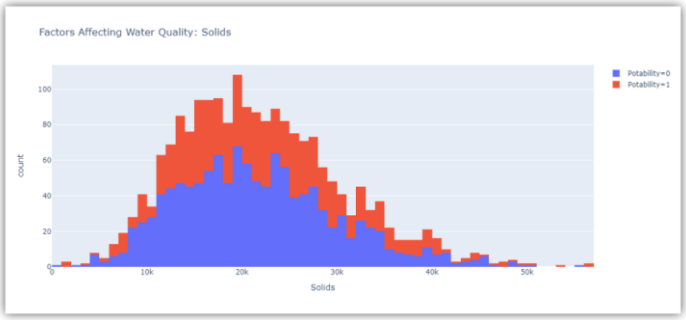
يمثل عمود الأس الهيدروجيني قيمة الأس الهيدروجيني للماء وهو عامل مهمفي تقييم التوازن الحمضي القاعدي للماء. يجب أن تكون قيمة الرقم الهيدروجيني لمياه الشرب بين 6.5 و8.5. دعونا الآن نلقي نظرة على العامل الثاني الذي يؤثر على جودة المياهفي مجموعة البيانات:

```
figure = px.histogram(data, x = "Hardness",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Hardness")
figure.show()
```



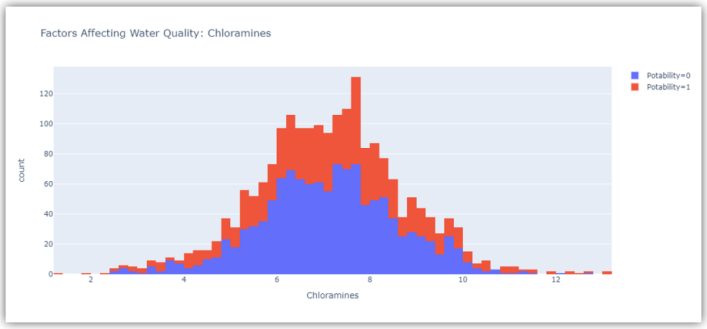
يوضح الشكل أعلاه توزيع عسرة hardness الماءفي مجموعة البيانات. تعتمد عسرة الماء عادة على مصدره، لكن الماء الذي تصل قوته إلى 120ـ200 ملليغرام صالح للشرب. دعنا الآن نلقي نظرة على العامل التالي الذي يؤثر على جودة المياه:

```
figure = px.histogram(data, x = "Solids",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Solids")
figure.show()
```



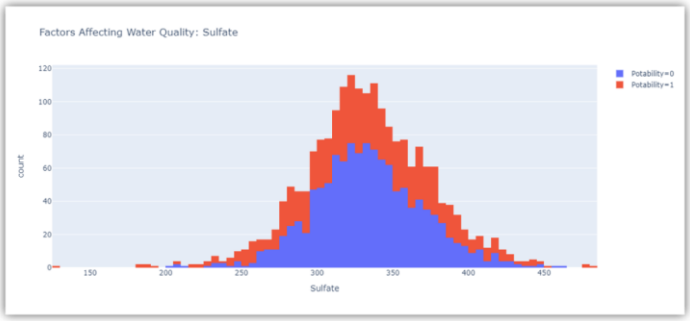
يمثل الشكل أعلاه توزيع إجمالي المواد الصلبة الذائبةفي الماءفي مجموعة البيانات. تسمى جميع المعادن العضوية وغير العضوية الموجودةفي الماء بالمواد الصلبة الذائبة. الماء الذي يحتوي على عدد كبير جداً من المواد الصلبة الذائبة شديد التمعدن. دعنا الآن نلقي نظرة على العامل التالي الذي يؤثر على جودة المياه:

```
figure = px.histogram(data, x = "Chloramines",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Chloramines")
figure.show()
```



يمثل الشكل أعلاه توزيع الكلورامين (chloramine)في الماءفي مجموعة البيانات. الكلورامين والكلور من المطهرات المستخدمةفي أنظمة المياه العامة. دعنا الآن نلقي نظرة على العامل التالي الذي يؤثر على جودة المياه:

```
figure = px.histogram(data, x = "Sulfate",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Sulfate")
figure.show()
```

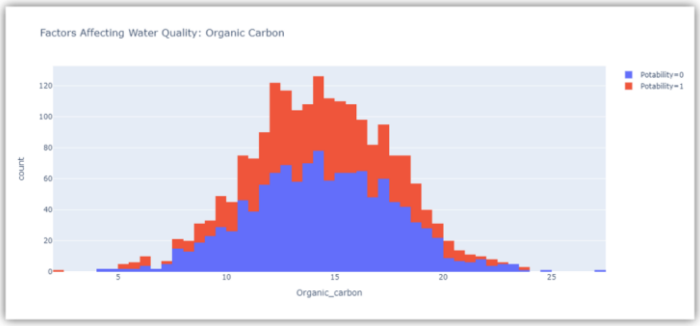


يوضح الشكل أعلاه توزيع الكبريتات (sulfate)في الماءفي مجموعة البيانات. إنها مواد موجودة بشكل طبيعيفي المعادن والتربة والصخور. الماء الذي يحتوي على أقل من 500 ملليغرام من الكبريتات آمن للشرب. الآن دعونا نرى العامل التالي:

```
figure = px.histogram(data, x = "Conductivity",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Conductivity")
figure.show()
```

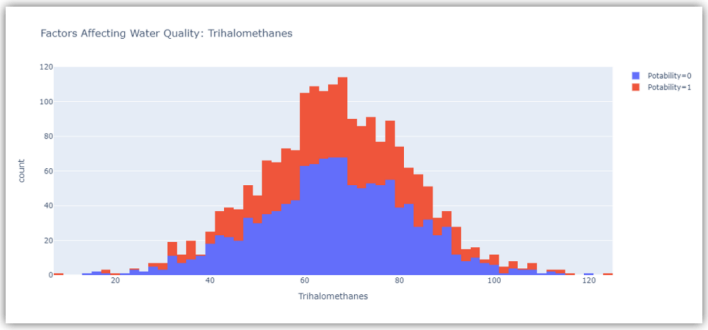
يمثل الشكل أعلاه توزيع موصلية المياه (water conductivity)في مجموعة البيانات. الماء موصل جيد للكهرباء، لكن أنقى أشكال الماء ليس موصلًا جيدًا للكهرباء. المياه ذات التوصيل الكهربائي أقل من 500 صالحة للشرب. الآن دعونا نرى العامل التالي:

```
figure = px.histogram(data, x = "Organic_carbon",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Organic Carbon")
figure.show()
```



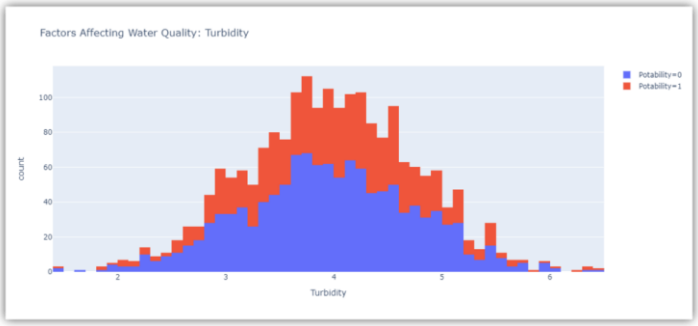
يمثل الشكل أعلاه توزيع الكربون العضوي (organic carbon)في الماءفي مجموعة البيانات. يأتي الكربون العضوي من انهيار المواد العضوية الطبيعية والمصادر الاصطناعية. تعتبر المياه التي تحتوي على أقل من 25 ملليجرام من الكربون العضوي آمنة للشرب. دعنا الآن نلقي نظرة على العامل التالي الذي يؤثر على جودة مياه الشرب:

```
figure = px.histogram(data, x = "Trihalomethanes",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Trihalomethanes")
figure.show()
```



يمثل الشكل أعلاه توزيع ثلاثي الميثان (trihalomethanes) أو (THM)في الماءفي مجموعة البيانات. THMs هي مواد كيميائية موجودةفي المياه المعالجة بالكلور. تعتبر المياه التي تحتوي على أقل من 80 ملليجرام من THMs آمنة للشرب. دعنا الآن نلقي نظرة على العامل التاليفي مجموعة البيانات الذي يؤثر على جودة مياه الشرب:

```
figure = px.histogram(data, x = "Turbidity",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Turbidity")
figure.show()
```



يمثل الشكل أعلاه توزيع العكارة (turbidity)في الماء. تعتمد عكارة الماء على عدد المواد الصلبة الموجودةفي المعلق. تعتبر المياه ذات العكارة أقل من 5 ملليغرام صالحة للشرب.

نموذج التنبؤ بجودة المياه باستخدام لغة بايثون

في القسم أعلاه، استكشفنا جميع الميزات التي تؤثر على جودة المياه. الآن، الخطوة التالية هي تدريب نموذج التعلم الآلي لمهمة تحليل جودة المياه باستخدام بايثون. لهذه المهمة، سأستخدم مكتبة PyCaretفي بايثون. إذا لم تكن قد استخدمت هذه المكتبة من قبل، فيمكنك تثبيتها بسهولة على نظامك باستخدام الأمر pip:

```
pip install pycaret
```

قبل تدريب نموذج التعلم الآلي، دعنا نلقي نظرة على الارتباط (correlation) بين جميع الميزات فيما يتعلق بعمود القابلية للشرب (Potability column)في مجموعة البيانات:

```
correlation = data.corr()
correlation["ph"].sort_values(ascending=False)
```

```
ph                1.000000
Hardness          0.108948
Organic_carbon    0.028375
Trihalomethanes   0.018278
Potability        0.014530
Conductivity      0.014128
Sulfate           0.010524
Chloramines      -0.024768
Turbidity        -0.035849
Solids           -0.087615
Name: ph, dtype: float64
```

الآن فيما يلي كيف يمكنك معرفة أي خوارزمية تعلم الآلة هي الأفضل لمجموعة البيانات هذه باستخدام مكتبة PyCaretفي بايثون:

```
from pycaret.classification import*
clf = setup(data, target = "Potability", silent = True,
session_id = 786)
compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.6830	0.7005	0.4197	0.6744	0.5133	0.2976	0.3182	0.724
qda	Quadratic Discriminant Analysis	0.6823	0.7192	0.3985	0.6883	0.5013	0.2917	0.3174	0.022
et	Extra Trees Classifier	0.6816	0.6941	0.3861	0.6858	0.4916	0.2863	0.3123	0.557
lightgbm	Light Gradient Boosting Machine	0.6652	0.6916	0.4762	0.6078	0.5324	0.2781	0.2840	0.172
gbc	Gradient Boosting Classifier	0.6602	0.6738	0.3718	0.6306	0.4667	0.2419	0.2603	0.339
nb	Naive Bayes	0.6184	0.6078	0.2478	0.5545	0.3412	0.1261	0.1462	0.019
dt	Decision Tree Classifier	0.6034	0.5895	0.5186	0.5049	0.5097	0.1775	0.1784	0.027
lr	Logistic Regression	0.5984	0.5199	0.0071	0.1900	0.0134	0.0028	0.0127	0.355
ridge	Ridge Classifier	0.5984	0.0000	0.0089	0.1583	0.0168	0.0035	0.0056	0.021
lda	Linear Discriminant Analysis	0.5977	0.4903	0.0089	0.1500	0.0167	0.0021	0.0024	0.022
ada	Ada Boost Classifier	0.5956	0.5671	0.2919	0.4896	0.3644	0.0972	0.1034	0.173
knn	K Neighbors Classifier	0.5743	0.5423	0.3644	0.4642	0.4070	0.0826	0.0846	0.121
svm	SVM - Linear Kernel	0.5194	0.0000	0.3982	0.1604	0.2287	-0.0014	-0.0104	0.027

وفقًا للنتيجة أعلاه، فإن خوارزمية الغابة العشوائية random forrest(rf) هي الأفضل لتدريب نموذج التعلم الآلي لمهمة تحليل جودة المياه. لذلك دعونا ندرب النموذج ونفحص تنبؤاته:

```
model = create_model("rf")
predict = predict_model(model, data=data)
predict.head()
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability	Label	Score
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0	0	0.87
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0	0	0.91
5	5.584087	188.313324	28748.687739	7.544869	326.678363	280.467916	8.399735	54.917862	2.559708	0	0	0.83
6	10.223862	248.071735	28749.716544	7.513408	393.663396	283.651634	13.789695	84.603556	2.672989	0	0	0.89
7	8.635849	203.361523	13672.091764	4.563009	303.309771	474.607645	12.363817	62.798309	4.401425	0	0	0.94

النتائج المذكورة أعلاه تبدو مرضية. آمل أن تكون قد أحببت مشروع التعلم الآلي هذا حول تحليل جودة المياه باستخدام بايثون.

الملخص

هذه هي الطريقة التي يمكنك بها تحليل جودة المياه وتدريب نموذج التعلم الآلي لتصنيف المياه الآمنة وغير الآمنة للشرب. يُعد الحصول على مياه الشرب المأمونة أحد الاحتياجات الأساسية لجميع البشر. من وجهة نظر قانونية، يعتبر الحصول على مياه الشرب أحد حقوق الإنسان الأساسية. تؤثر العديد من العوامل على جودة المياه، كما أنها أحد مجالات البحث الرئيسيةفي التعلم الآلي.

46) التنبؤ بمبيعات ألعاب الفيديو باستخدام التعلم الآلي Video Game Sales Prediction with Machine Learning

تحليل مبيعات ألعاب الفيديو (Video game sales analysis) هو بيان مشكلة شائع على Kaggle. يمكنك العمل على هذه المشكلة لتحليل مبيعات أكثر من 16500 لعبة أو يمكنك أيضًا تدريب نموذج التعلم الآلي للتنبؤ بمبيعات ألعاب الفيديو. لذلك إذا كنت تريد معرفة كيفية تدريب نموذج تنبؤ بمبيعات ألعاب الفيديو، فهذه المقالة مناسبة لك.في هذه المقالة، سأوجهك خلال مهمة تعلم الآلة حول تدريب نموذج توقع مبيعات لعبة فيديو باستخدام بايثون.

نموذج التنبؤ بمبيعات ألعاب الفيديو باستخدام بايثون

تحليل بيانات المبيعات لأكثر من 16500 لعبة هو بيان مشكلة شائع جدًا على Kaggle. يمكنك إما حل هذه المشكلة للعثور على العديد من الأنماط (patterns) والعلاقات (relationships) بين العوامل التي تؤثر على مبيعات ألعاب الفيديو، أو يمكنك استخدام مجموعة البيانات هذه للتنبؤ بمبيعات ألعاب الفيديوفي المستقبل. لذلكفي القسم أدناه، سأوجهك خلال كيفية تدريب نموذج التعلم الآلي للتنبؤ بمبيعات ألعاب الفيديو باستخدام بايثون.

تحتوي مجموعة البيانات التي أستخدمها لهذه المهمة على قائمة بألعاب الفيديو ومبيعاتها. لنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv("vgsales.csv")
print(data.head())
```

	Rank	Name	Platform	Year	...	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	...	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	...	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	...	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	...	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	...	8.89	10.22	1.00	31.37

دعنا الآن نرى ما إذا كانت مجموعة البيانات هذه تحتوي على قيم خالية:

```
print(data.isnull().sum())
```

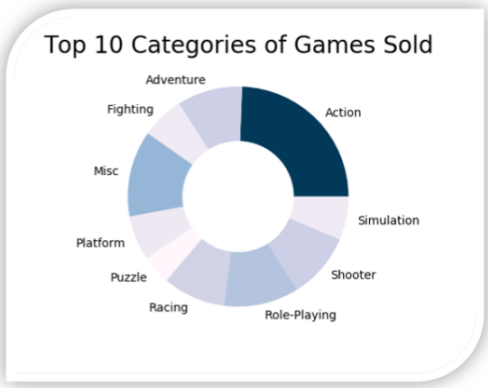
```
Rank            0
Name            0
Platform        0
Year          271
Genre           0
Publisher      58
NA_Sales        0
EU_Sales        0
JP_Sales        0
Other_Sales     0
Global_Sales    0
dtype: int64
```

سأقوم الآن بإنشاء مجموعة بيانات جديدة تزيل القيم الخالية:

```
data = data.dropna()
```

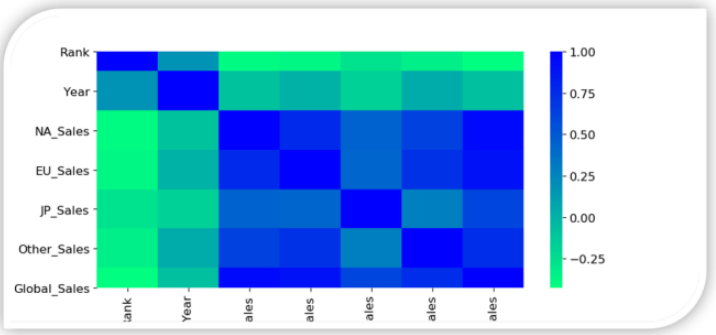
قبل تدريب النموذج، دعنا نلقي نظرة على أفضل 10 فئات من الألعاب مبيعًا:

```
import matplotlib as mpl
game = data.groupby("Genre")["Global_Sales"].count().head(10)
custom_colors = mpl.colors.Normalize(vmin=min(game),
vmax=max(game))
colours = [mpl.cm.PuBu(custom_colors(i)) for i in game]
plt.figure(figsize=(7,7))
plt.pie(game, labels=game.index, colors=colours)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Top 10 Categories of Games Sold", fontsize=20)
plt.show()
```



دعنا الآن نلقي نظرة على الارتباط (correlation) بين ميزات مجموعة البيانات هذه:

```
print(data.corr())
sns.heatmap(data.corr(), cmap="winter_r")
plt.show()
```



نموذج التنبؤ بمبيعات ألعاب الفيديو التدريبية

دعنا الآن نرى كيفية تدريب نموذج التعلم الآلي للتنبؤ بمبيعات ألعاب الفيديو باستخدام بايثون . سأقوم بإعداد البيانات عن طريق تخزين الميزات التي نحتاجها لتدريب هذا النموذجفي المتغير xوتخزين العمود الهدفي المتغيرy :

```
x = data[["Rank", "NA_Sales", "EU_Sales", "JP_Sales",
"Other_Sales"]]
y = data["Global_Sales"]
```

الآن دعنا نقسم البيانات ونستخدم خوارزمية الانحدار الخطي (linear regression) لتدريب هذا النموذج:

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.2, random_state=42)

from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(xtrain, ytrain)
predictions = model.predict(xtest)
```

الملخص

هذه هي الطريقة التي يمكننا بها تدريب نموذج التعلم الآلي للتنبؤ بمبيعات ألعاب الفيديو. هذا بيان مشهور لمشكلة Kaggle يمكنك استخدامه لتحسين مهاراتكفي العمل مع البيانات والتدريب على نموذج التعلم الآلي. آمل أن تكون قد أحببت هذه المقالة حول كيفية تدريب نموذج التنبؤ بمبيعات ألعاب الفيديو باستخدام بايثون.

المصادر

1. 125 Data Science Projects You Can Try with Python, Aman Kharwal, https://python.plainenglish.io/85-data-science-projects-c03c8750599e.
2. 40+ Data Analysis Projects with Python, Aman Kharwal, https://amankharwal.medium.com/data-analysis-projects-with-python-a262a6f9e68c.

Data Science & Machine Learning

45 Data Science Projects with ML Solved and Explained with Python

By: Dr. Alaa Taima