السلاسل الزمنية

(التحليل والتنبؤ)

عن طريق الامثلة

٢٥ مشروع سلسلة زمنية تم حلها وشرحها باستخدام بايثون

ترجمة واعداد: د. علاء طعيمة

بـــسم تعالى

الـــلاـــل الزمنية: عن طريق الامثلة

25 مشروع سلسلة زمنية تم حلها وشرحها باستخدام بايثون

ترجمة واعداد:

د. علاء طعيمة

المقدمة

بيانات السلاسل الزمنية time series هي سلسلة من البيانات التي يتم جمعها خلال فترة زمنية. عندما نعمل على بيانات السلاسل الزمنية لتحليل الأنماط والعثور عليها، يُعرف ذلك باسم تحليل السلاسل الزمنية والتنبؤ بها Time Series Analysis and forecasting. لذلك، إذا كنت تريد أن تصبح عالم بيانات، فإن تحليل السلاسل الزمنية والتنبؤ بها هو أحد الموضوعات المهمة بالنسبة لك. إذا كنت تبحث عن بعض من أفضل أفكار مشاريع تعلم الآلة وعلم البيانات في تحليل السلاسل الزمنية والتنبؤ بها، فهذا الكتاب مناسب لك.في هذه الكتاب، سوف تتعرف على بعض من أفضل المشاريع حول تحليل السلاسل الزمنية والتنبؤ بها التي يجب أن تجربها.

لقد حاولت قدر المستطاع ان اترجم المشاريع الأكثر طرحاًفي مجال تحليل السلاسل الزمنية والتنبؤ بها 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 مجال تحليل والتنبؤ بالسلاسل الزمنية ومساعدة القارئ العربي على تعلم هذا المجال. اسأل الله التوفيقفي هذا العمل لأثراء المحتوى العربي الذي يفتقر أشد الافتقار إلى محتوى جيد ورصينفي مجال تحليل السلاسل الزمنية والتنبؤ بها. ونرجو لك الاستمتاع مع الكتاب ولا تنسونا من صالح الدعاء.

د. علاء طعيمة

كلية علوم الحاسوب وتكنولوجيا المعلومات

جامعة القادسية

العراق

المحتويات

0) أفضل الطرق لتحليل السلاسل الزمنية Best Approaches for Time Series Analysis

يعد تحليل السلاسل الزمنية Time series analysis أحد أهم الموضوعاتفي علم البيانات. بيانات السلاسل الزمنية Time series data هي سلسلة من نقاط البيانات التي تم جمعها وفهرستها بناءً على فترة زمنية، وعندما نقوم بتحليل هذه البيانات للعثور على أنماط خلال فترة زمنية، يُعرف ذلك باسم تحليل السلاسل الزمنية. إذا كنت تريد معرفة أفضل الطرق لتحليل السلاسل الزمنية التي تحتاج إلى معرفتها، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أقدم لكم بعضًا من أفضل الطرق لحل مشاكل تحليل السلاسل الزمنية.

أفضل الطرق لتحليل السلاسل الزمنية

ARIMA

ARIMA هي خوارزمية شهيرة للتنبؤ بالسلسلة الزمنية ، وهي اختصار لـ المتوسط المتحرك الانحدار التلقائي (Autoregressive Integrated Moving Average). تتنبأ هذه الخوارزمية بقيمة وفقًا للجمع الخطي للبيانات التاريخية لمجموعة بيانات السلاسل الزمنية. يعد استخدام ARIMA طريقة قوية ومرنة لتحليل السلاسل الزمنية. يمكنك معرفة المزيد حول استخدام ARIMA لتحليل السلاسل الزمنية من هنا.

SARIMA

SARIMA هي خوارزمية أخرى شهيرة للتنبؤ بالسلاسل الزمنية، وهي اختصار لـ المتوسط المتحرك الموسمي ذاتي الانحدار الذاتي (Seasonal Autoregressive Integrated Moving Average) أو ARIMA الموسمي (Seasonal ARIMA). إذا كانت بيانات السلاسل الزمنية الخاصة بك تحتوي على أنماط موسمية (أحد المكونات المهمة لنمذجة السلاسل الزمنية الخاصة بك)، فإن خوارزمية SARIMA مفضلة على خوارزمية ARIMA. يمكنك معرفة المزيد حول استخدام SARIMA لتحليل السلاسل الزمنية من هنا.

LSTM

LSTM هي بُنية شبكة عصبية ، تعني شبكة ذاكرة طويلة قصيرة المدى (Long Short Term Memory network). إنه نوع من الشبكات العصبية المتكررة recurrent neural network يُفضل عندما تحتاج إلى نموذجك لتذكر البيانات لفترة طويلة. يُستخدم LSTMفي التنبؤ بالسلاسل الزمنية والعديد من المشكلات الأخرى بناءً على تحليل الانحدار regression analysis. يمكنك معرفة تطبيق LSTM من هنا.

نموذج Facebook Prophet

يتم استخدام نموذج Facebook Prophet في التنبؤ بالسلسلة الزمنية بناءً على مجموعة بيانات ذات اتجاهات غير خطية مع تأثيرات موسمية وأسبوعية وحتى يومية. إنه نموذج تسلسل زمني تلقائي تم إنشاؤه بواسطة مطوري Facebook. إذا كنت تستخدم R أو بايثون لعلم البيانات، فيمكنك استخدام هذا النموذج. يمكنك معرفة المزيد عن نموذج Facebook Prophet لتحليل السلاسل الزمنية من هنا.

AutoTS

AutoTS هي مكتبة تلقائية للتعلم الآلي في بايثون، تم تطويرها للتنبؤ التلقائي بالسلاسل الزمنية. في الغالب، أفضل هذه المكتبة للتنبؤ بأسعار الأسهم stock prices والعملات المشفرة cryptocurrencies. يمكنك العثور على تطبيقه باستخدام بايثون من هنا.

الملخص

بيانات السلاسل الزمنية هي سلسلة من نقاط البيانات التي تم جمعها وفهرستها وفقًا لفاصل زمني، وعندما نقوم بتحليل هذه البيانات للعثور على أنماط على مدى فترة زمنية، يُعرف ذلك باسم تحليل السلاسل الزمنية. أتمنى أن تكون قد أحببت هذا المقال الذي يتناول أفضل الطرق لتحليل السلاسل الزمنية.

المصدر:

https://thecleverprogrammer.com/2022/01/08/best-approaches-for-/time-series-analysis

1) تحليل السلاسل الزمنية باستخدام بايثون Time Series Analysis using Python

يعني تحليل السلاسل الزمنية Time series analysis تحليل وإيجاد الأنماطفي مجموعة بيانات السلاسل الزمنية time-series dataset. مجموعة بيانات السلاسل الزمنية هي سلسلة من البيانات التي يتم جمعها خلال فترة زمنية. تعد بيانات أسعار الأسهم وبيانات المبيعات الشهرية وبيانات هطول الأمطار اليومية وبيانات حركة مرور موقع الويب بالساعة بعض الأمثلة على بيانات السلاسل الزمنية التي ستحصل عليها لحل مشكلات العمل بصفتك عالم بيانات. لذلك إذا كنت تريد معرفة تحليل السلاسل الزمنية، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة تحليل السلاسل الزمنية باستخدام بايثون.

تحليل السلاسل الزمنية

عندما تقوم بتحليل مجموعة بيانات مسجلة خلال فترة زمنية، فأنت تقوم بتحليل السلاسل الزمنية. يمكن أن يكون الفاصل الزمني لبيانات السلاسل الزمنية أسبوعيًا أو شهريًا أو يوميًا أو حتى كل ساعة، لكن عملية تحليل بياناتك ستظل كما هيفي معظم المشكلات.

في نهاية هذه المقالة، ستتعلم إجراء تحليل السلاسل الزمنية باستخدام بايثون. سأستخدم مكتبة plotlyفي بايثون هنا لأنه من السهل تحليل البيانات بشكل مخطط بسبب قلة الكود والنتائج التفاعلية. سأوصي باستخدام Jupyter Notebook أو Google Colaboratory لتحليل السلاسل الزمنية بدلاً من استخدام محرر كود أو IDE مثل VS Code أو PyCharm.

تحليل السلاسل الزمنية باستخدام بايثون

لنبدأ مهمة تحليل السلاسل الزمنية باستخدام بايثون عن طريق استيراد مكتبات بايثون الضرورية ومجموعة بيانات السلاسل الزمنية:

```
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=720)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2

data = yf.download('AAPL',
          start=start_date,
          end=end_date,
          progress=False)
```

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

Date	Open	High	Low	Close	Adj Close	Volume
2020-01-28	78.150002	79.599998	78.047501	79.422501	78.260017	162234000
2020-01-29	81.112503	81.962502	80.345001	81.084999	79.898186	216229200
2020-01-30	80.135002	81.022499	79.687500	80.967499	79.782402	126743200
2020-01-31	80.232498	80.669998	77.072502	77.377502	76.244957	199588400
2020-02-03	76.074997	78.372498	75.555000	77.165001	76.035568	173788400

في الكود أعلاه، استخدمت yfinance API لاستخراج أحدث بيانات أسعار الأسهم. يمكنك معرفة المزيد عنها من هنا. الآن دعنا نرسم مخطط خط لمعرفة الاتجاهاتفي أسعار أسهم Apple:

```
import plotly.express as px
figure = px.line(data, x = data.index,
       y = "Close",
       title = "Time Series Analysis (Line Plot)")
figure.show()
```

المخطط الخطي line plot هو أحد أفضل أدوات التصوير أثناء العمل على تحليل السلاسل الزمنية.في الكود أعلاه، أتخيل الاتجاهاتفي أسعار إغلاق Apple. إذا وضعت المؤشر على السطر، فسترى سعر الإغلاقفي التاريخ المحدد لنقطة البيانات التي يوجد بها المؤشر.

الآن دعنا نرسم مخطط الشموع candlestick لرؤية الاتجاهاتفي أسعار الفتح والعالية والمنخفضة والإغلاق لشركة Apple:

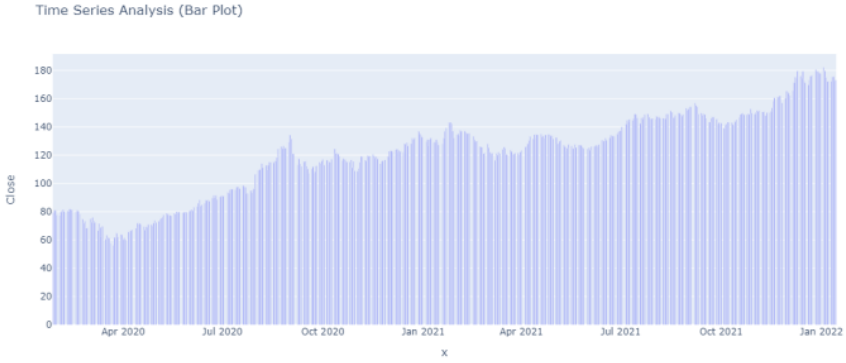
```
import plotly.graph_objects as go
figure = go.Figure(data=[go.Candlestick(x = data.index,
                open = data["Open"],
                high = data["High"],
                low = data["Low"],
                close = data["Close"])])
```

```
figure.update_layout(title = "Time Series Analysis
(Candlestick Chart)”,
        xaxis_rangeslider_visible = False)
figure.show()
```

يعد مخطط الشموع مفيدًا دائمًافي تحليل السلاسل الزمنية للأداة المالية. إذا قمت بوضع المؤشر على أي نقطةفي مخطط الشموع أعلاه، فسترى جميع أسعار Apple (مفتوح، مرتفع، منخفض، مغلق)في التاريخ الذي يوجد فيه المؤشر. تشير الخطوط الحمراءفي هذا المخطط إلى انخفاض الأسعار، بينما تشير الخطوط الخضراء إلى ارتفاع الأسعار.

الآن دعنا نرسم مخطط شريط لتصوير اتجاهات أسعار الإغلاق خلال الفترة:

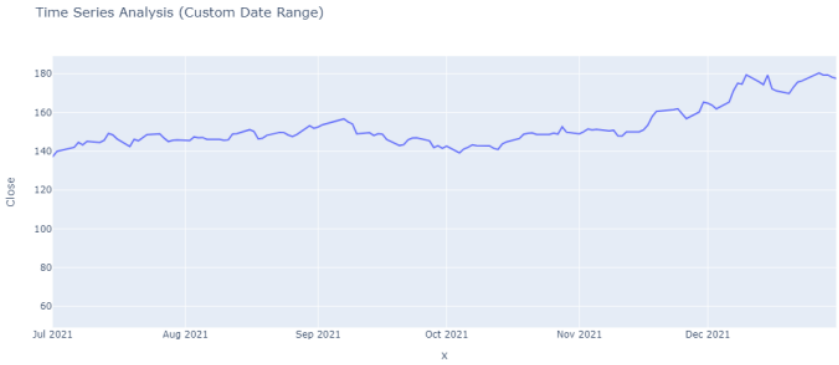
```
figure = px.bar(data, x = data.index,
      y = "Close”,
      title = "Time Series Analysis (Bar Plot)”)
figure.show()
```



يُظهر المخطط الشريطي bar plot أعلاه زيادةفي أسعار الأسهمفي سيناريو المدى الطويل. يوضح لك الرسم البياني الخطي ومخطط الشموع زيادة السعر وانخفاضه، ولكن إذا كنت تريد أن ترى السعر يرتفع وينخفض على المدى الطويل، فيجب عليك دائمًا تفضيل المخطط الشريطي.

إذا كنت ترغبفي تحليل أسعار الأسهم بين فترة تاريخين محددين، فيما يلي كيفية القيام بذلك:

```
figure = px.line(data, x = data.index,
       y = 'Close”,
       range_x,['31_12_2021','01_07_2021'] =
       title = "Time Series Analysis (Custom Date Range)”)
figure.show()
```




```
figure = go.Figure(data = [go.Candlestick(x = data.index,
                open = data["Open"],
                high = data["High"],
                low = data["Low"],
                close = data["Close"])]
figure.update_layout(title = "Time Series Analysis
(Candlestick Chart with Buttons and Slider)")

figure.update_xaxes(
  rangeslider_visible = True,
  rangeselector = dict(
    buttons = list([
      dict(count = 1, label = "1m", step = "month", stepmode =
"backward"),
      dict(count = 6, label = "6m", step = "month", stepmode =
"backward"),
      dict(count = 1, label = "YTD", step = "year", stepmode =
"todate"),
      dict(count = 1, label = "1y", step = "year", stepmode =
"backward"),
      dict(step = "all")
    ])
  )
)
figure.show()
```

إذن هذه هي الطريقة التي يمكنك بها إجراء تحليل السلاسل الزمنية باستخدام بايثون.

الملخص

أتمنى أن تكون قد فهمت الآن كيفية إجراء تحليل السلاسل الزمنية باستخدام بايثون وجميع التصويرات التي يمكنك استخدامها لتحليل السلاسل الزمنية. مجموعة بيانات السلاسل الزمنية هي سلسلة من البيانات التي يتم جمعها خلال فترة زمنية. يعني تحليل السلاسل الزمنية تحليل وإيجاد الأنماطفي مجموعة بيانات السلاسل الزمنية. يمكن أن يكون الفاصل الزمني لبيانات السلاسل الزمنية أسبوعيًا أو شهريًا أو يوميًا أو حتى كل ساعة. آمل أن تكون قد أحببت هذه المقالة حول تحليل السلاسل الزمنية باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2022/01/17/time-series-analysis-/using-python

2) كيفية اختيار نموذج التنبؤ بالسلاسل الزمنية How to Choose a Time Series Forecasting Model

التنبؤ بالسلاسل الزمنية Time Series Forecasting هو عملية تحليل ونمذجة بيانات السلاسل الزمنية. يساعدفي التنبؤ بالسلوك المستقبلي للسوق، وهو أمر مفيدفي صنع القرار لكل عمل تجاري. بعض تطبيقات التنبؤ بالسلاسل الزمنية هي التنبؤ بالطقس والمناخ، والتنبؤ بالمبيعات، والتنبؤ بالأعمال التجارية، والتنبؤ بسوق الأوراق المالية، وما إلى ذلك. أثناء العمل على مشكلة التنبؤ بالسلسلة الزمنية، يجب أن تعرف كيفية اختيار نموذج التنبؤ forecasting model. لذلك، إذا كنت لا تعرف كيفية اختيار نموذج التنبؤ، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذكفي جولة حول كيفية اختيار نموذج تنبؤ السلاسل الزمنية.

كيفية اختيار نموذج توقع السلاسل الزمنية

يعتمد اختيار نموذج التنبؤ بالسلاسل الزمنية على نوع بيانات السلاسل الزمنية التي تعمل بها. هناك أربعة أنواع من بيانات السلاسل الزمنية:

1. البيانات الثابتة Stationary data.
2. البيانات مع الاتجاهات Data with trends.
3. البيانات الموسمية Seasonal data.
4. بيانات السلاسل الدورية Cyclical series data.

لنستعرض كل هذه الأنواع من بيانات السلاسل الزمنية ونفهم كيفية اختيار نموذج التنبؤ بالسلاسل الزمنية لكل نوع من أنواع البيانات.

عندما تكون البيانات ثابتة

البيانات الثابتة Stationary data هي البيانات التي لا تتغير قيمتها المتوسطة على المدى الطويل. لنفترض المبيعات الشهرية للحليبفي المدينة. سيشتري مستهلكو الحليب الحليب يوميًا، كما سيشتري الباعة الحليب وفقًا للطلب على الحليبفي منطقتهم. إنه مثال على البيانات الثابتة حيث لا تتغير القيمة المتوسطة مع التغييرفي الوقت.

يعد نموذج (ARMA) Autoregressive Movie Average أحد أفضل النماذج التي يجب عليك اختيارها أثناء العمل على البيانات الثابتة.

عندما تكون البيانات مع الاتجاهات

توجد الاتجاهاتفي البيانات عندما تكون هناك زيادة طويلة الأجل أو نقصانفي مجموعة البيانات. يمكن أن تكون الزيادة أو النقصان خطية أو غير خطية. بكلمات بسيطة، التغييراتفي الأنماط

ليست ثابتة مع التغييرفي الوقت المناسب. على سبيل المثال، الطلب على الكهرباءفي منطقة معينة، وتكلفة إنتاج منتج أثناء التغيرات الاقتصادية، وما إلى ذلك.

يعد Autoregressive Integrated Moving Average model (ARIMA) أحد أفضل نماذج التنبؤ بالسلاسل الزمنية التي يجب عليك اختيارها أثناء العمل على البيانات ذات الاتجاهات.

عندما تكون البيانات موسمية

البيانات الموسمية Seasonal data هي البيانات التي تتأثر بالعوامل الموسمية. يمكن أن تكون العوامل الموسمية شهرًا معينًا من العام أو أسبوعًا من الشهر أو يومًا من أيام الأسبوع. بكلمات بسيطة، عندما تتكرر أنماط البيانات بعد وقت معين، تكون البيانات موسمية. على سبيل المثال، الحصول على وصول أعلىفي Instagram كل يوم أحد، ومبيعات عالية خلال موسم الأعياد، وما إلى ذلك.

يعد Seasonal Autoregressive Integrated Moving Average model (SARIMA) أحد أفضل نماذج التنبؤ بالسلاسل الزمنية التي يجب عليك اختيارها أثناء العمل على البيانات الموسمية.

عندما تكون بيانات السلاسل الدورية

بيانات السلاسل الدورية cyclical series data هي البيانات التي تحتوي على اتجاهات الزيادة والنقصان بدون تكرار ثابت. هذه الأنماط ترجع إلى سلوك السوق وبيئة الأعمال. بكلمات بسيطة، إذا كانت البيانات بها تقلبات بدون تكرار ثابت، فهي دورية. على سبيل المثال، سوق المنافسة المثالية، والتغيراتفي الموضة، والكوارث الطبيعية، إلخ.

يصعب التنبؤ بالسلسلة الدورية لأن الأنماط غير مستقرة. يتطلب التنبؤ ببيانات السلاسل الزمنية بسلسلة دورية إيجاد المؤشرات الاقتصادية الرائدة.

الملخص

يعتمد اختيار نموذج توقع السلاسل الزمنية على نوع بيانات السلاسل الزمنية التي تعمل بها. لذلك عندما تكون البيانات ثابتة، اختر نموذج ARMA. عندما تحتوي البيانات على اتجاهات، اختر نموذج ARIMA. عندما يكون موسميًا، اختر طراز SARIMA. وإذا كانت مجموعة البيانات تحتوي على سلسلة دورية، فأنت بحاجة إلى العثور على المؤشرات الاقتصادية الرائدة. آمل أن تكون قد أحببت هذه المقالة حول اختيار نموذج تنبؤ السلاسل الزمنية.

https://thecleverprogrammer.com/2022/06/23/heres-how-to-choose-a-/time-series-forecasting-model

3) تحليل سوق أسهم تويتر باستخدام لغة بايثون Twitter Stock Market Analysis using Python

Twitter هو أحد تطبيقات الوسائط الاجتماعية الشائعة حيث يشارك الأشخاص ما يشعرون به في عدد محدود من الكلمات. Twitter مشهور ولكن ليس في سوق الأسهم. نظرًا لأن Twitter غير مدرج في بورصة نيويورك، فلنحلل الجدول الزمني الكامل لتويتر في سوق الأسهم. في هذه المقالة، سأطلعك على مهمة تحليل سوق أسهم تويتر باستخدام بايثون.

تحليل سوق الأسهم على تويتر باستخدام لغة بايثون

بدأ Twitter رحلته في سوق الأسهم في عام 2013. لذا لتحليل الجدول الزمني الكامل لتويتر في سوق الأسهم، نحتاج إلى بيانات أسعار أسهم تويتر من 2013 إلى 2022. لقد وجدت مجموعة بيانات تحتوي على البيانات التي نحتاجها لهذه المهمة. يمكنك تنزيل مجموعة البيانات من هنا.

لنبدأ الآن بمهمة تحليل سوق أسهم Twitter من خلال استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import datetime
from datetime import date, timedelta
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
pio.templates.default = "plotly_white"

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

```
         Date       Open       High        Low      Close  Adj Close  \
0  2013-11-07  45.099998  50.090000  44.000000  44.900002  44.900002
1  2013-11-08  45.930000  46.939999  40.685001  41.650002  41.650002
2  2013-11-11  40.500000  43.000000  39.400002  42.900002  42.900002
3  2013-11-12  43.660000  43.779999  41.830002  41.900002  41.900002
4  2013-11-13  41.029999  42.869999  40.759998  42.599998  42.599998

      Volume
0  117701670.0
1   27925307.0
2   16113941.0
3    6316755.0
4    8688325.0
```

تحتوي مجموعة البيانات على بيانات حول:

1. التاريخ Date.

2. سعر الافتتاح اليوم The opening Price of the day.
3. أعلى سعرفي اليوم The highest price of the day.
4. أقل سعرفي اليوم The lowest price of the day.
5. سعر اغلاق اليوم The closing price of the day.
6. سعر الإغلاق المعدل لليوم The adjusted closing price of the day.
7. إجمالي عدد الأسهم المتداولةفي اليوم (الحجم) The total number of shares traded in the day (volume).

دعونا نلقي نظرة على إحصاءات العمود:

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2264 entries, 0 to 2263
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       2264 non-null   object
 1   Open       2259 non-null   float64
 2   High       2259 non-null   float64
 3   Low        2259 non-null   float64
 4   Close      2259 non-null   float64
 5   Adj Close  2259 non-null   float64
 6   Volume     2259 non-null   float64
dtypes: float64(6), object(1)
memory usage: 123.9+ KB
None
```

عمود التاريخ Date column هو كائنفي مجموعة البيانات هذه. سنقوم بتحويله إلى نوع بيانات التاريخ والوقت لاحقًا. الآن، دعنا نلقي نظرة على ما إذاكانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

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

```
Date         0
Open         5
High         5
Low          5
Close        5
Adj Close    5
Volume       5
dtype: int64
```

توجد خمس قيم فارغةفي كل عمود باستثناء عمود التاريخ. دعنا نزيل الصفوف ذات القيم الخالية وننتقل إلى أبعد من ذلك:

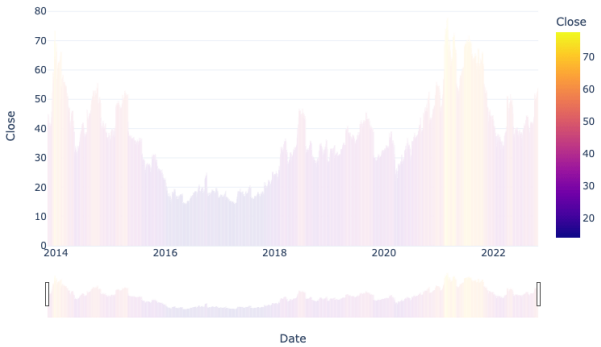
```
data = data.dropna()
```

دعنا الآن نلقي نظرة على أسعار أسهم Twitter على مر السنين:

```
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 = "Twitter Stock Prices Over the
Years ,"
                     xaxis_rangeslider_visible=False)
figure.show()
```

لذلك منذ إدخال Twitterفي سوق الأسهم، كان مربحًا فقطفي بداية و2021. دعنا نتخيل مخططًا شريطيًا لتحليل أسعار أسهم Twitter بالتفصيل:

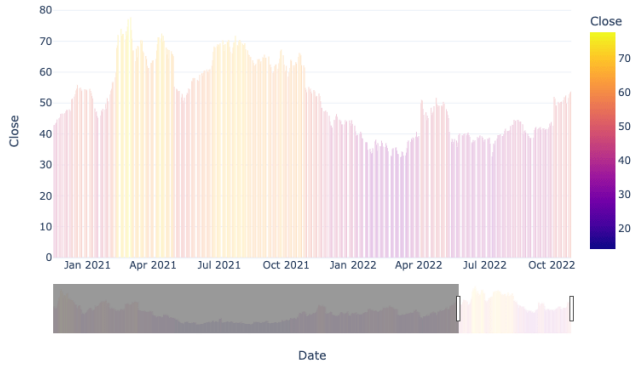
```
figure = px.bar(data,
       x = "Date”,
       y= "Close”,
       color="Close("
figure.update_xaxes(rangeslider_visible=True)
figure.show()
```



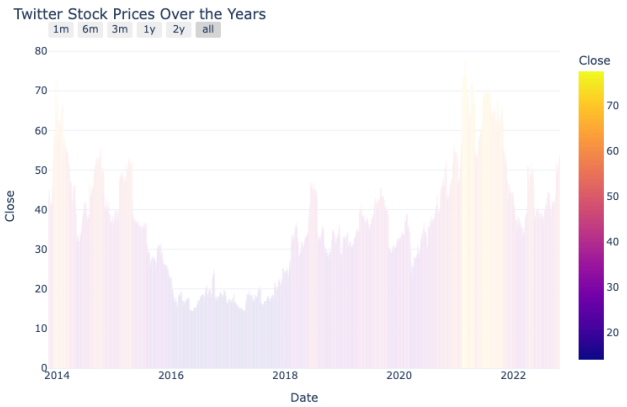
يوضح الرسم البياني أعلاه أسعار أسهم Twitter على مر السنين. يمكنك استخدام شريط تمرير النطاق range slider أدناه لتكبير فترة زمنية معينة. انظر إلى الصورة كمثال.

باستخدام شريط تمرير النطاق، يمكننا أن نرى أن الربع الأول من عام 2021 كان أفضل فترة زمنية لتويتر على مدار السنواتفي سوق الأسهم. يمكننا أيضًا تخصيص أزرار للتحكمفي الفترات الزمنية. دعنا نضيف أزرار لتحليل أسعار أسهم Twitterفي فترات زمنية مختلفة:

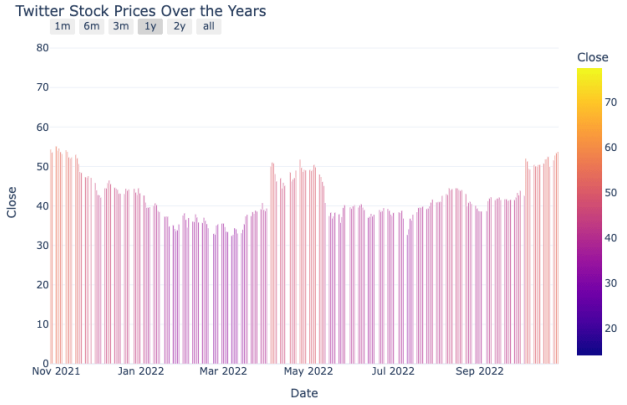
```
figure = px.bar(data, x = "Date", y= "Close", color="Close")
figure.update_xaxes(rangeslider_visible=True)
figure.update_layout(title = "Twitter Stock Prices Over the
Years ,"
        xaxis_rangeslider_visible=False)
figure.update_xaxes(
  rangeselector=dict(
    buttons=list([
      dict(count=1, label="1m", step="month",
stepmode="backward"),
      dict(count=6, label="6m", step="month",
stepmode="backward"),
      dict(count=3, label="3m", step="month",
```




```
stepmode="backward"),
      dict(count=1, label="1y", step="year",
stepmode="backward"),
      dict(count=2, label="2y", step="year",
stepmode="backward"),
      dict(step="all")
    ])
  )
)
figure.show()
```



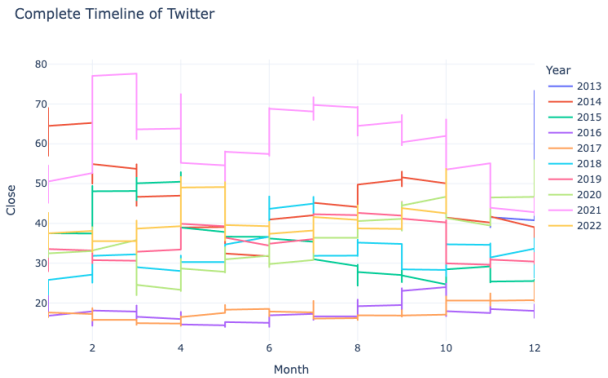
ستساعدك الأزرار الموجودةفي التصوير أعلاه على فهم أسعار أسهم Twitterفي فترات زمنية مختلفة. انظر إلى الصورة أدناه كمثال.



عندما نقرنا على الزر "1y"، يظهر الرسم البياني أداء Twitterفي سوق الأسهم خلال العام الماضي.

دعنا الآن نلقي نظرة على الجدول الزمني الكامل لـ Twitterفي سوق الأسهم:

```
data["Date"] = pd.to_datetime(data["Date”],
               format = '%Y-%m-%d’)
data['Year'] = data['Date'].dt.year
data["Month"] = data["Date"].dt.month
fig = px.line(data,
      x="Month”,
      y="Close”,
      color='Year’,
      title="Complete Timeline of Twitter”)
fig.show()
```



لذلك، منذ إدخال Twitter في سوق الأسهم، سار 2014 بشكل جيد بالنسبة إلى Twitter في السنوات الأربع الأولى. كان عامي 2016 و2017 هما الأسوأ بالنسبة لتويترفي سوق الأسهم. ارتفعت أسعار أسهمهافي 2018 و2019 و2020. ثم جاء عام 2021، أفضل عام لتويترفي سوق الأسهم. وصل Twitter إلى أعلى سعر سهم له على الإطلاقفي عام 2021. لكن أسعار أسهم Twitter انخفضت مرة أخرىفي عام 2022.

الملخص

هذه هي الطريقة التي يمكنك من خلالها تحليل الجدول الزمني الكامل لتويترفي سوق الأسهم من 2013 إلى 2022. تويتر هو أحد تطبيقات التواصل الاجتماعي الشهيرة ولا يزال يزداد شعبية بعد أن استحوذ Elon Musk على Twitter. لكنها لم تكن أبدًا من بين أفضل الشركات أداءًفي سوق الأسهم. آمل أن تكون قد أحببت هذا المقال على تحليل سوق أسهم Twitter باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2022/11/07/twitter-stock-market-/analysis-using-python

4) التنبؤ بالطقس باستخدام بايثون Weather Forecasting using Python

في علم البيانات (Data Science)، التنبؤ بالطقس هو تطبيق للتنبؤ بالسلاسل الزمنية (Time Series Forecasting) حيث نستخدم بيانات وخوارزميات السلاسل الزمنية لعمل تنبؤات لوقت معين. إذا كنت تريد معرفة كيفية التنبؤ بالطقس باستخدام مهاراتكفي علم البيانات، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بالطقس (weather forecasting) باستخدام بايثون.

التنبؤ بالطقس

التنبؤ بالطقس هو مهمة التنبؤ بأحوال الطقس لموقع ووقت معين. باستخدام بيانات وخوارزميات الطقس، من الممكن التنبؤ بأحوال الطقس لعدد n من الأيام القادمة.

للتنبؤ بالطقس باستخدام بايثون، نحتاج إلى مجموعة بيانات تحتوي على بيانات الطقس التاريخية بناءً على موقع معين. لقد وجدت مجموعة بيانات على Kaggle استنادًا إلى بيانات الطقس اليوميةفي نيودلهي. يمكننا استخدام مجموعة البيانات هذه لمهمة التنبؤ بالطقس. يمكنك تنزيل مجموعة البيانات من هنا.

في القسم أدناه، ستتعلم كيف يمكننا تحليل الطقس والتنبؤ به باستخدام بايثون.

تحليل بيانات الطقس باستخدام بايثون

الآن لنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات التي نحتاجها:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

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

	date	meantemp	humidity	wind_speed	meanpressure
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000

دعونا نلقي نظرة على الإحصائيات الوصفية لهذه البيانات قبل المضي قدمًا:

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

```
          meantemp     humidity   wind_speed  meanpressure
count  1462.000000  1462.000000  1462.000000   1462.000000
mean     25.495521    60.771702     6.802209   1011.104548
std       7.348103    16.769652     4.561602    180.231668
min       6.000000    13.428571     0.000000     -3.041667
25%      18.857143    50.375000     3.475000   1001.580357
50%      27.714286    62.625000     6.221667   1008.563492
75%      31.305804    72.218750     9.238235   1014.944901
max      38.714286   100.000000    42.220000   7679.333333
```

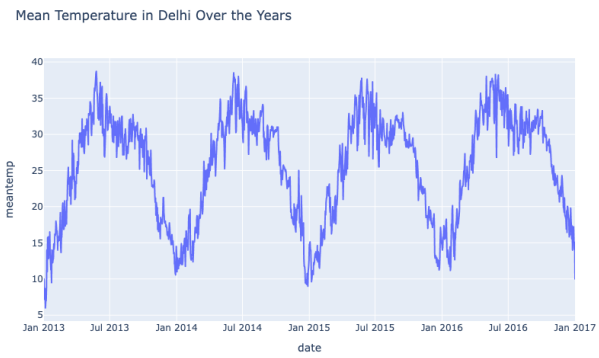
دعنا الآن نلقي نظرة على المعلومات المتعلقة بجميع الأعمدةفي مجموعة البيانات:

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1462 entries, 0 to 1461
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   date          1462 non-null   object 
 1   meantemp      1462 non-null   float64
 2   humidity      1462 non-null   float64
 3   wind_speed    1462 non-null   float64
 4   meanpressure  1462 non-null   float64
dtypes: float64(4), object(1)
memory usage: 57.2+ KB
```

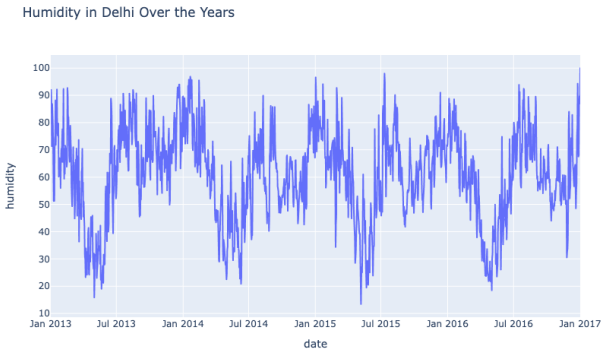
لا يحتوي عمود التاريخ (date column)في مجموعة البيانات هذه على نوع بيانات التاريخ والوقت. سنقوم بتغييره عند الحاجة. دعونا نلقي نظرة على متوسط درجة الحرارة (mean temperature)في دلهي على مر السنين:

```
figure = px.line(data, x="date",
                 y="meantemp",
                 title='Mean Temperature in Delhi Over the
Years')
figure.show()
```



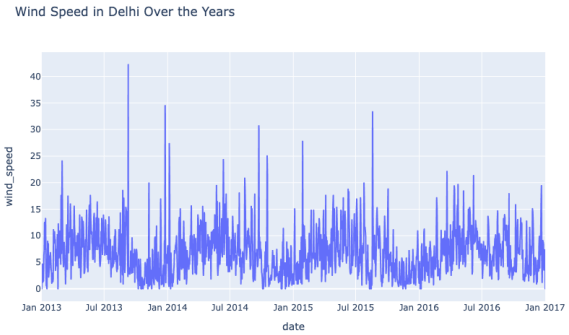
الآن دعونا نلقي نظرة على الرطوبة (humidity)في دلهي على مر السنين:

```
figure = px.line(data, x="date ,"
                 y="humidity ,"
                 title='Humidity in Delhi Over the Years('
figure.show()
```



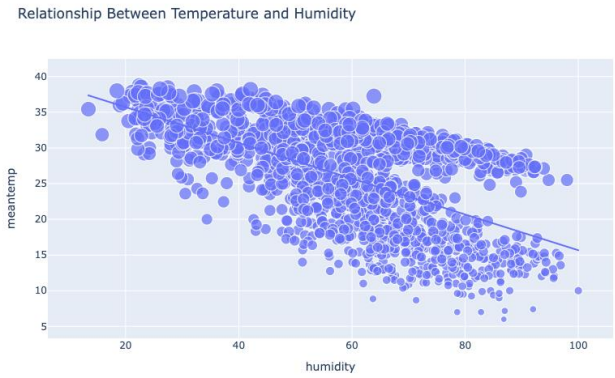
الآن دعونا نلقي نظرة على سرعة الرياح (wind speed)في دلهي على مر السنين:

```
figure = px.line(data, x="date ,"
                 y="wind_speed ,"
                 title='Wind Speed in Delhi Over the Years('
figure.show()
```



حتى عام 2015، كانت سرعة الرياح أعلى خلال الرياح الموسمية (أغسطس وسبتمبر) وتراجع الرياح الموسمية (ديسمبر ويناير). بعد عام 2015، لم تكن هناك حالات شاذةفي سرعة الرياح خلال الرياح الموسمية. دعنا الآن نلقي نظرة على العلاقة بين درجة الحرارة والرطوبة:

```
figure = px.scatter(data_frame = data, x="humidity,"
                    y="meantemp", size="meantemp ,"
                    trendline="ols ,"
                    title = "Relationship Between Temperature
and Humidity("
figure.show()
```



هناك علاقة سلبية بين درجة الحرارة والرطوبةفي دلهي. وهذا يعني أن ارتفاع درجة الحرارة يؤدي إلى انخفاض الرطوبة وانخفاض درجة الحرارة يؤدي إلى ارتفاع نسبة الرطوبة.

تحليل تغير درجة الحرارة

الآن دعونا نحلل التغيرفي درجة الحرارةفي دلهي على مر السنين. بالنسبة لهذه المهمة، سأقوم أولاً بتحويل نوع بيانات عمود التاريخ إلى التاريخ والوقت. ثم سأضيف عمودين جديدينفي مجموعة البيانات لقيم السنة والشهر.

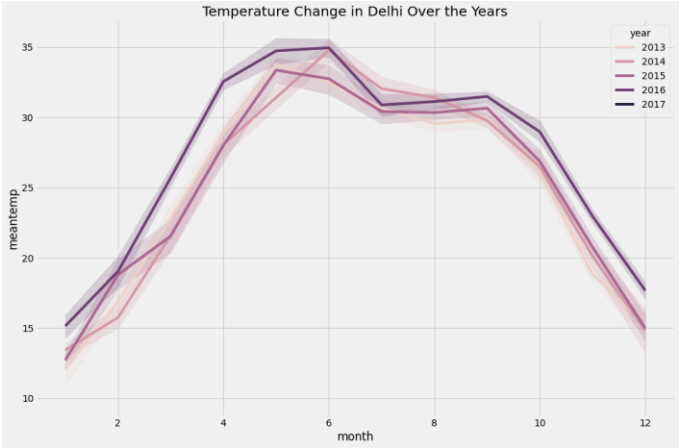
إليك كيفية تغيير نوع البيانات واستخراج بيانات السنة والشهر من عمود التاريخ:

```
data["date"] = pd.to_datetime(data["date"], format = '%Y-%m-
%d')
data['year'] = data['date'].dt.year
data["month"] = data["date"].dt.month
print(data.head())
```

	date	meantemp	humidity	wind_speed	meanpressure	year	month
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667	2013	1
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000	2013	1
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667	2013	1
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667	2013	1
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000	2013	1

الآن دعونا نلقي نظرة على تغير درجة الحرارةفي دلهي على مر السنين:

```
plt.style.use('fivethirtyeight')
plt.figure(figsize=(15, 10))
plt.title("Temperature Change in Delhi Over the Years")
sns.lineplot(data = data, x='month', y='meantemp', hue='year')
plt.show()
```



على الرغم من أن عام 2017 لم يكن الأكثر سخونةفي الصيف، يمكننا أن نرى ارتفاعًافي متوسط درجة حرارة دلهي كل عام.

التنبؤ بالطقس باستخدام بايثون

الآن دعنا ننتقل إلى مهمة التنبؤ بالطقس. سأستخدم نموذج (Facebook prophet) لهذه المهمة. يعد نموذج Facebook prophet أحد أفضل التقنيات للتنبؤ بالسلاسل الزمنية. إذا لم تستخدم هذا النموذج من قبل، فيمكنك تثبيته على نظامك باستخدام الأمر المذكور أدناهفي موجه الأوامر أو التيرمينال:

```
pip install prophet
```

يقبل نموذج prophet بيانات الوقت المسماة "ds" والتسميات على أنها "y". فلنحول البيانات إلى هذا التنسيق:

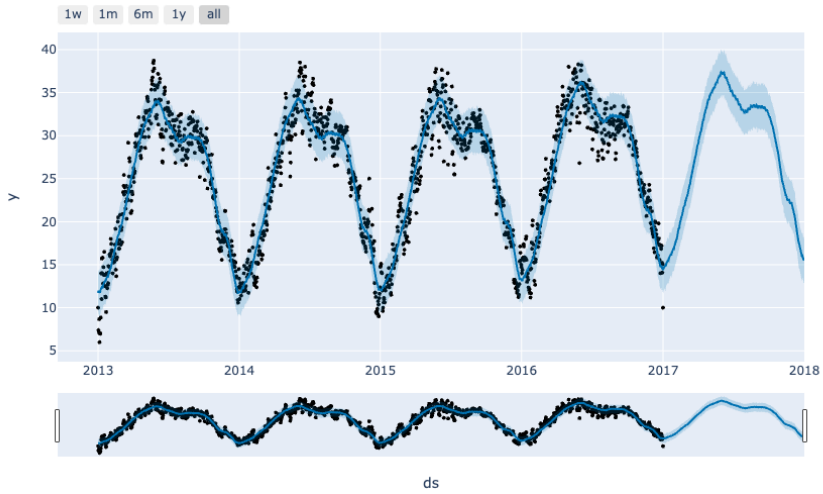
```
forecast_data = data.rename(columns = {"date": "ds",
"                                      meantemp": "y"})
print(forecast_data)
```

	ds	y	humidity	wind_speed	meanpressure	year	month
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667	2013	1
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000	2013	1
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667	2013	1
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667	2013	1
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000	2013	1
...	...	...	...	...	...	...	...
1457	2016-12-28	17.217391	68.043478	3.547826	1015.565217	2016	12
1458	2016-12-29	15.238095	87.857143	6.000000	1016.904762	2016	12
1459	2016-12-30	14.095238	89.666667	6.266667	1017.904762	2016	12
1460	2016-12-31	15.052632	87.000000	7.325000	1016.100000	2016	12
1461	2017-01-01	10.000000	100.000000	0.000000	1016.000000	2017	1

[1462 rows x 7 columns]

الآن فيما يلي كيفية استخدام نموذج Facebook prophet للتنبؤ بالطقس باستخدام بايثون:

```
from prophet import Prophet
from prophet.plot import plot_plotly, plot_components_plotly
model = Prophet()
model.fit(forecast_data)
forecasts = model.make_future_dataframe(periods=365)
predictions = model.predict(forecasts)
plot_plotly(model, predictions)
```



هذه هي الطريقة التي يمكنك من خلالها تحليل الطقس والتنبؤ به باستخدام بايثون.

الملخص

التنبؤ بالطقس هو مهمة التنبؤ بأحوال الطقس لموقع ووقت معين. باستخدام بيانات وخوارزميات الطقس، من الممكن التنبؤ بأحوال الطقس لعدد n من الأيام القادمة. أتمنى أن تكون قد أحببت هذه المقالة حول تحليل الطقس والتنبؤ باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2022/10/17/weather-forecasting-/using-python

5) التنبؤ بحالات Covid-19 باستخدام بايثون Covid-19 Cases Prediction with Python

في هذه المقالة، سأقدم لك مشروع التعلم الآلي حول التنبؤ بحالات Covid-19 باستخدام بايثون للأيام الثلاثين القادمة. تساعد هذه الأنواع من النماذج التنبؤيةفي توفير تنبؤ دقيق بالأوبئة، وهو أمر ضروري للحصول على معلومات حول الانتشار المحتمل للأمراض المعدية وعواقبها.

تعتمد الحكومات والهيئات التشريعية الأخرى على هذه الأنواع من النماذج والأفكار التنبؤية للتعلم الآلي لاقتراح سياسات جديدة وتقييم فعالية السياسات المطبقة.

مشروع التعلم الآلي على التنبؤ بحالات Covid-19 باستخدام بايثون

سأبدأ مهمة التنبؤ بحالات Covid-19 باستخدام بايثون للأيام الثلاثين القادمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

تحميل مجموع البيانات 1

تحميل مجموع البيانات 2

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

from fbprophet import Prophet
from sklearn.metrics import r2_score

plt.style.use("ggplot")

df0 = pd.read_csv("CONVENIENT_global_confirmed_cases.csv")
df1 = pd.read_csv("CONVENIENT_global_deaths.csv")
```

تحضير البيانات

الآن الخطوة التالية هي إعداد البيانات data preparation، سأقوم ببساطة بإعداد بيانات جديدة من خلال دمج مجموعات البيانات المذكورة أعلاه ثم سنقوم بتصوير مخطط جغرافي geographical plot للبيانات لمعرفة ما سنعمل معه:

```
world = pd.DataFrame({"Country":[],"Cases":[]})
world["Country"] = df0.iloc[:,1:].columns
cases = []
for i in world["Country"]:
    cases.append(pd.to_numeric(df0[i][1:]).sum())
world["Cases"]=cases

country_list=list(world["Country"].values)
idx = 0
for i in country_list:
```

```
    sayac = 0
    for j in i:
        if j==".":
            i = i[:sayac]
            country_list[idx]=i
        elif j=="(":
            i = i[:sayac-1]
            country_list[idx]=i
        else:
            sayac += 1
    idx += 1
world["Country"]=country_list
world = world.groupby("Country")["Cases"].sum().reset_index()
world.head()
continent=pd.read_csv("continents2.csv")
continent["name"]=continent["name"].str.upper()
```

	Country	Cases
0	Afghanistan	45716.0
1	Albania	35600.0
2	Algeria	79110.0
3	Andorra	6534.0
4	Angola	14920.0

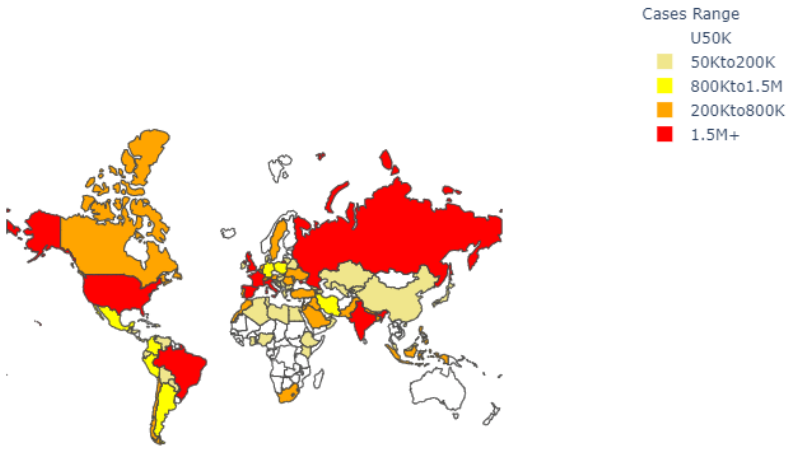
العرض المرئي للبيانات

الآن هنا سأقوم بإعداد ثلاثة تصويرات. سيكون أحدها تصويرًا جغرافيًا لتصور الانتشار العالمي لـ Covid-19. ثم سيكون التصوير التالي هو إلقاء نظرة على الحالات اليومية لـ Covid-19 في العالم. ثم سيكون التصوير الأخير هو إلقاء نظرة على حالات الوفاة اليومية لـ Covid-19في العالم.

الآن دعنا نبدأ تصوير البيانات من خلال النظرفي الانتشار العالمي لـ Covid-19:

```
world["Cases Range"]=pd.cut(world["Cases"],[-
150000,50000,200000,800000,1500000,15000000],labels=["U50K","5
0Kto200K","200Kto800K","800Kto1.5M","1.5M+"])
alpha =[]
for i in world["Country"].str.upper().values:
    if i == "BRUNEI":
        i="BRUNEI DARUSSALAM"
    elif  i=="US":
        i="UNITED STATES"
    if len(continent[continent["name"]==i]["alpha-
3"].values)==0:
        alpha.append(np.nan)
```

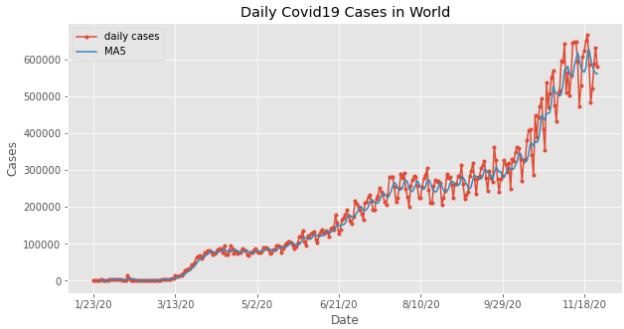
```
    else:
        alpha.append(continent[continent["name"]==i]["alpha-
3"].values[0])
world["Alpha3"]=alpha

fig = px.choropleth(world.dropna(),
                    locations="Alpha3",
                    color="Cases Range",
                     projection="mercator",

color_discrete_sequence=["white","khaki","yellow","orange","re
d"])
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



الآن دعونا نلقي نظرة على الحالات اليوميةفي جميع أنحاء العالم:

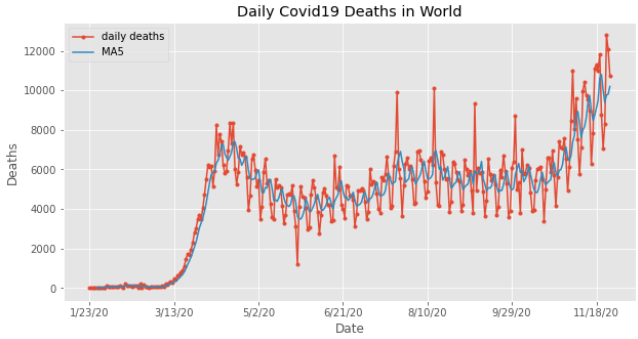
```
count = []
for i in range(1,len(df0)):
    count.append(sum(pd.to_numeric(df0.iloc[i,1:].values)))

df = pd.DataFrame()
df["Date"] = df0["Country/Region"][1:]
df["Cases"] = count
df=df.set_index("Date")

count = []
for i in range(1,len(df1)):
    count.append(sum(pd.to_numeric(df1.iloc[i,1:].values)))

df["Deaths"] = count

df.Cases.plot(title="Daily Covid19 Cases in
World",marker=".",figsize=(10,5),label="daily cases")
df.Cases.rolling(window=5).mean().plot(figsize=(10,5),label="M
A5")
plt.ylabel("Cases")
plt.legend()
```

```
plt.show()
```



الآن دعونا نلقي نظرة على حالات الوفاة اليومية لـ Covid-19:

```
df.Deaths.plot(title="Daily Covid19 Deaths in
World",marker=".",figsize=(10,5),label="daily deaths")
df.Deaths.rolling(window=5).mean().plot(figsize=(10,5),label="
MA5")
plt.ylabel("Deaths")
plt.legend()
plt.show()
```



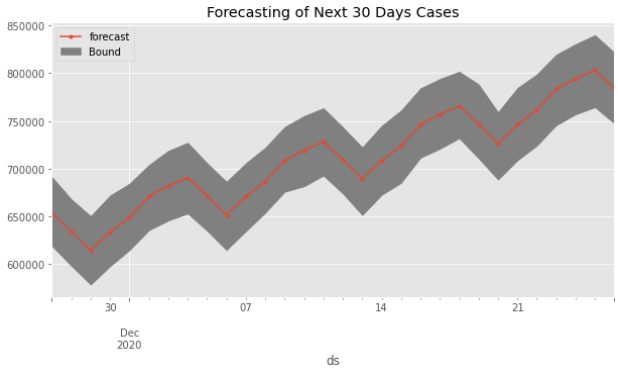
التنبؤ بحالات Covid-19 باستخدام بايثون للأيام الثلاثين القادمة

الآن، سأستخدم نموذج Facebook prophet لمهمة التنبؤ بحالات Covid-19 مع بايثون للأيام الثلاثين القادمة. يستخدم نموذج Facebook prophet طريقة السلاسل الزمنية للتنبؤ.

دعونا نرى كيف يمكننا استخدام نموذج Facebook prophet للتنبؤ بحالات Covid-19 مع بايثون للأيام الثلاثين القادمة:

```
class Fbprophet(object):
    def fit(self,data):

        self.data  = data
        self.model =
Prophet(weekly_seasonality=True,daily_seasonality=False,yearly
_seasonality=False)
        self.model.fit(self.data)

    def forecast(self,periods,freq):
```

```
        self.future =
self.model.make_future_dataframe(periods=periods,freq=freq)
        self.df_forecast = self.model.predict(self.future)

    def plot(self,xlabel="Years",ylabel="Values"):


self.model.plot(self.df_forecast,xlabel=xlabel,ylabel=ylabel,f
igsize=(9,4))

self.model.plot_components(self.df_forecast,figsize=(9,6))

    def R2(self):
        return r2_score(self.data.y,
self.df_forecast.yhat[:len(df)])

df_fb  = pd.DataFrame({"ds":[],"y":[]})
df_fb["ds"] = pd.to_datetime(df.index)
df_fb["y"]  = df.iloc[:,0].values
model = Fbprophet()
model.fit(df_fb)
model.forecast(30,"D")
model.R2()
forecast =
model.df_forecast[["ds","yhat_lower","yhat_upper","yhat"]].tai
l(30).reset_index().set_index("ds").drop("index",axis=1)
forecast["yhat"].plot(marker=".",figsize=(10,5))
plt.fill_between(x=forecast.index, y1=forecast["yhat_lower"],
y2=forecast["yhat_upper"],color="gray")
plt.legend(["forecast","Bound"],loc="upper left")
plt.title("Forecasting of Next 30 Days Cases")
plt.show()
```



آمل أن تكون قد أحببت هذه المقالة حول تنبؤات حالات Covid-19 للأيام الثلاثين القادمة باستخدام لغة برمجة بايثون.

https://thecleverprogrammer.com/2020/11/29/covid-19-cases-/prediction-with-python

6) 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:

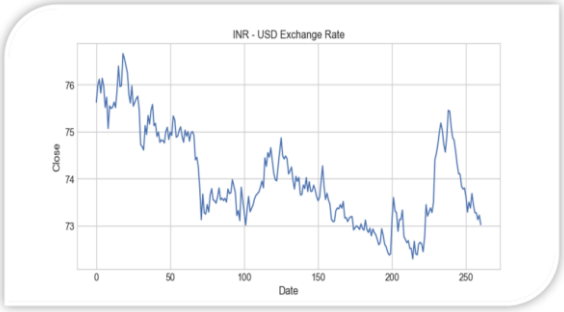
```
import numpy as np
import pandas as pd
```

```
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:

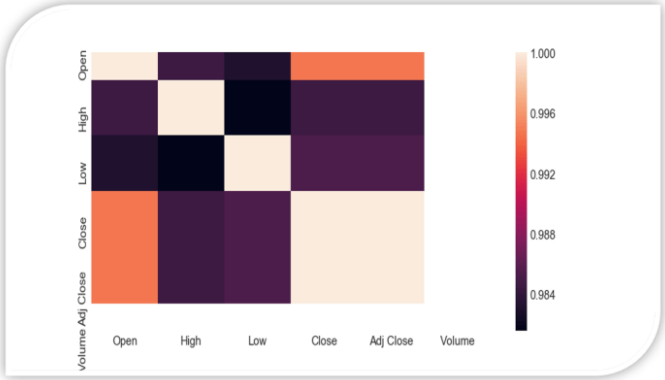
```
plt.figure(figsize=(10, 4))
plt.title("INR - USD Exchange Rate")
plt.xlabel("Date")
plt.ylabel("Close")
plt.plot(data["Close"])
plt.show()
```



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.

المصدر:

https://thecleverprogrammer.com/2021/05/22/currency-exchange-rate-/prediction-with-machine-learning

7) التنبؤ بترافيك موقع ويب باستخدام بايثون Website Traffic Forecasting using Python

يعني التنبؤ بترافيك موقع الويب (Website Traffic Forecasting) التنبؤ بترافيك موقع الويب خلال فترة معينة. إنها واحدة من أفضل حالات استخدام تنبؤ السلاسل الزمنية (Time Series Forecasting). إذا كنت تريد معرفة كيفية التنبؤ بترافيك موقع ويب، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بترافيك موقع ويب باستخدام بايثون.

التنبؤ بترافيك موقع ويب باستخدام بايثون

يتم جمع مجموعة البيانات التي أستخدمها للتنبؤ بترافيك موقع الويب من بيانات الترافيك اليومية لموقع thecleverprogrammer.com. يحتوي على بيانات حول بيانات الترافيك اليومية من يونيو 2021 إلى يونيو 2022. يمكنك تنزيل مجموعة البيانات من هنا. لنبدأ الآنفي مهمة التنبؤ بترافيك موقع الويب عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.arima_model import ARIMA
import statsmodels.api as sm

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

```
         Date  Views
0  01/06/2021   7831
1  02/06/2021   7798
2  03/06/2021   7401
3  04/06/2021   7054
4  05/06/2021   7973
```

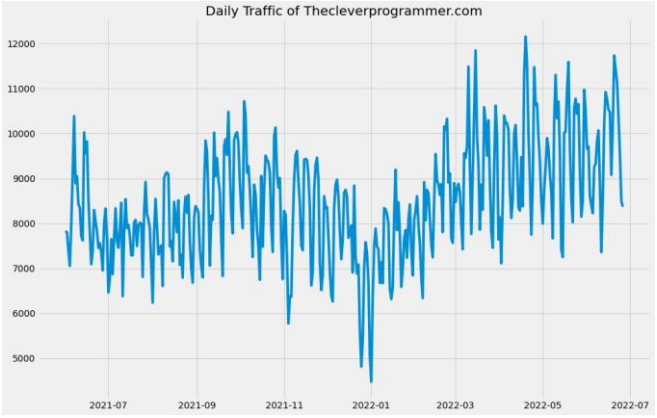
تحتوي مجموعة البيانات على عمودين، التاريخ (date) والترافيك(traffic). قبل المضي قدمًا، سأحول عمود التاريخ إلى نوع بيانات التاريخ والوقت:

```
data["Date"] = pd.to_datetime(data["Date"],
                              format="%d/%m/%Y")
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 391 entries, 0 to 390
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    391 non-null    datetime64[ns]
 1   Views   391 non-null    int64
dtypes: datetime64[ns](1), int64(1)
memory usage: 6.2 KB
None
```

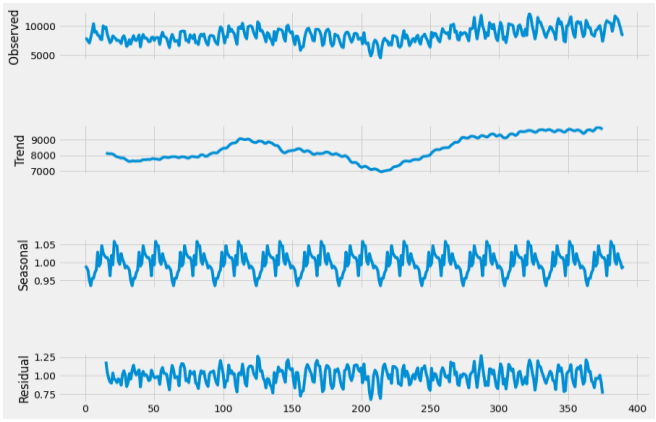
كان عمود التاريخ والوقت كائنًا (object)في البداية، لذلك قمت بتحويله إلى عمود التاريخ والوقت. دعنا الآن نلقي نظرة على الترافيك اليومية لموقع الويب:

```
plt.style.use('fivethirtyeight')
plt.figure(figsize=(15, 10))
plt.plot(data["Date"], data["Views"])
plt.title("Daily Traffic of Thecleverprogrammer.com")
plt.show()
```



بيانات الترافيك على موقعنا موسمية (seasonal) لأن الترافيك على الموقع تزداد خلال أيام الأسبوع وتنخفض خلال عطلات نهاية الأسبوع. من المفيد معرفة ما إذا كانت مجموعة البيانات موسمية أم لا أثناء العمل على مشكلة تنبؤ السلاسل الزمنية. فيما يلي كيف يمكننا إلقاء نظرة على ما إذا كانت مجموعة البيانات الخاصة بنا ثابتة (stationary) أو موسمية(seasonal):

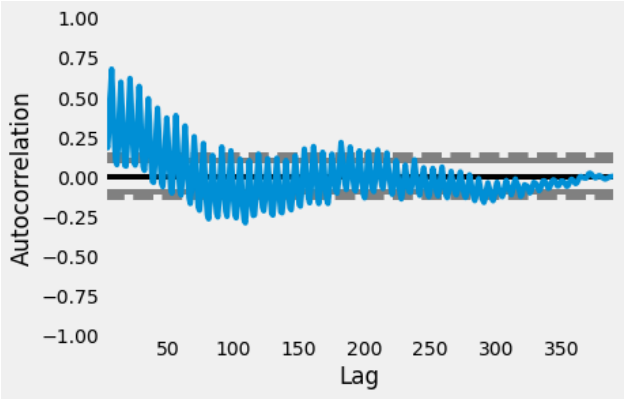
```
result = seasonal_decompose(data["Views"],
            model='multiplicative',
            freq = 30)
fig = plt.figure ()
fig = result.plot ()
fig.set_size_inches(10,15)
```



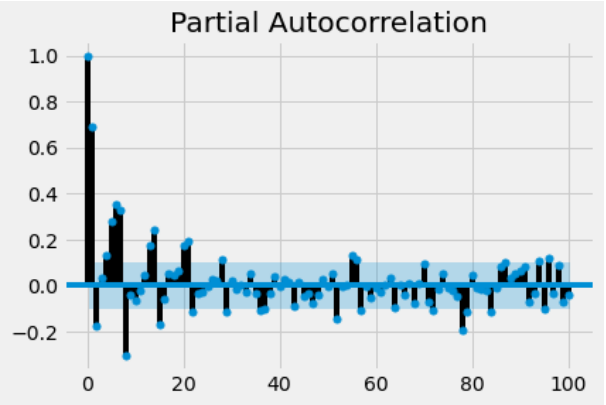
سأستخدم نموذج ARIMA الموسمي (SARIMA) للتنبؤ بالترافيك على موقع الويب. قبل استخدام نموذج SARIMA، من الضروري إيجاد قيم p و d و q. يمكنك معرفة كيفية العثور على قيم p و d و q من هنا.

نظرًا لأن البيانات ليست ثابتة، فإن قيمة d هي 1. للعثور على قيم p و q، يمكننا استخدام مخططات الارتباط التلقائي (autocorrelation) والارتباط التلقائي الجزئي (partial autocorrelation):

```
pd.plotting.autocorrelation_plot(data["Views"])
```




```
plot_pacf(data["Views"], lags = 100)
```



الآن إليك كيف يمكننا تدريب نموذج SARIMA لمهمة التنبؤ بترافيك موقع الويب:

```
p, d, q = 5, 1, 2
model=sm.tsa.statespace.SARIMAX(data['Views"],
                order=(p, d, q),
                seasonal_order=(p, d, q, 12))
model=model.fit()
print(model.summary())
```

```
                                Statespace Model Results
==========================================================================================
Dep. Variable:                             Views   No. Observations:                  391
Model:             SARIMAX(5, 1, 2)x(5, 1, 2, 12)   Log Likelihood               -3099.402
Date:                            Tue, 28 Jun 2022   AIC                           6228.803
Time:                                    07:01:10   BIC                           6287.827
Sample:                                         0   HQIC                          6252.229
                                              - 391
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.7808      0.134      5.836      0.000       0.519       1.043
ar.L2         -0.7973      0.135     -5.920      0.000      -1.061      -0.533
ar.L3         -0.1442      0.170     -0.850      0.395      -0.477       0.188
ar.L4         -0.1833      0.151     -1.210      0.226      -0.480       0.114
ar.L5         -0.1548      0.139     -1.117      0.264      -0.426       0.117
ma.L1         -1.1826      0.094    -12.515      0.000      -1.368      -0.997
ma.L2          0.8856      0.078     11.304      0.000       0.732       1.039
ar.S.L12      -0.2606      4.608     -0.057      0.955      -9.293       8.772
ar.S.L24       0.0428      0.781      0.055      0.956      -1.488       1.573
ar.S.L36      -0.1880      0.246     -0.764      0.445      -0.670       0.294
ar.S.L48      -0.2151      0.959     -0.224      0.823      -2.095       1.664
ar.S.L60       0.0127      0.986      0.013      0.990      -1.920       1.946
ma.S.L12      -0.6902      4.611     -0.150      0.881      -9.728       8.348
ma.S.L24      -0.0994      3.637     -0.027      0.978      -7.228       7.029
sigma2      1.257e+06   1.59e+05      7.914      0.000    9.46e+05    1.57e+06
===================================================================================
Ljung-Box (Q):                      102.98   Jarque-Bera (JB):                 1.32
Prob(Q):                              0.00   Prob(JB):                         0.52
Heteroskedasticity (H):               1.03   Skew:                             0.14
Prob(H) (two-sided):                  0.85   Kurtosis:                         3.01
===================================================================================
```

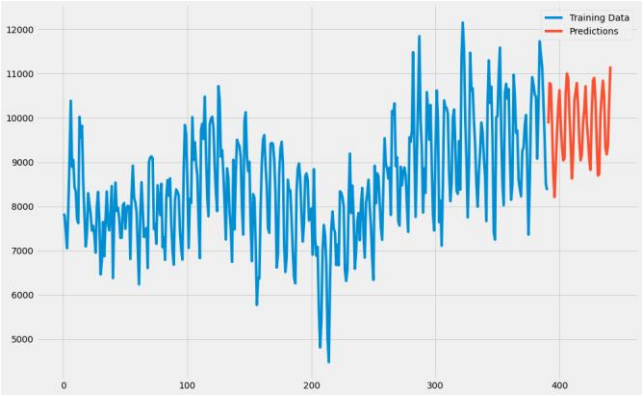
دعنا الآن نتنبأ بترافيك موقع الويب خلال الخمسين يومًا القادمة:

```
predictions = model.predict(len(data), len(data)+50)
print(predictions)
```

```
391     9874.390136
392    10786.957398
393    10757.445305
394     9863.890552
395     8765.031698
396     8212.310651
397     8929.181869
398     9685.809771
399    10270.622236
400    10625.904093
401     9854.870630
402     9362.193417
403     9040.021193
404     9081.558484
405    10538.993124
406    11003.816870
407    10897.859601
408    10083.291284
409     9445.806523
410     8629.901288
411     9184.420361
412    10392.770399
413    10593.941868
414    10788.128238
415    10263.101427
416     9449.467789
417     9040.226113
418     9168.972091
419     9887.094079
420    10218.658067
421    10715.657122
422     9899.224399
423     9541.622897
424     9065.810941
425     8825.335634
426    10137.936392
427    10839.866240
428    10905.862922
429    10411.640309
430     9451.211368
431     8698.339931
432     8725.534103
433    10060.678587
434    10506.263524
435    10842.515622
436    10485.387495
437     9335.244813
438     9175.122336
439     9357.034382
440    10295.910655
441    11162.934817
dtype: float64
```

إليك كيف يمكننا رسم التنبؤات:

```
data["Views"].plot(legend=True, label="Training Data”,
        figsize=(15, 10))
predictions.plot(legend=True, label="Predictions")
```



الملخص

إذن هذه هي الطريقة التي يمكنك بها التنبؤ بترافيك موقع الويب لفترة معينة. يعد التنبؤ بترافيك موقع الويب أحد أفضل أفكار مشروع علم البيانات التي يمكنك ذكرهافي سيرتك الذاتية. آمل أن يكون هذا المقال مفيداً لك لتعلم التنبؤ بترافيك موقع الويب باستخدام لغة برمجة بايثون.

المصدر:

https://thecleverprogrammer.com/2022/06/28/website-traffic-/forecasting-using-python

8) 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.

المصدر:

https://thecleverprogrammer.com/2022/01/03/stock-price-prediction-/with-lstm

9) التنبؤ بالمبيعات باستخدام التعلم الآلي Sales Forecasting with Machine Learning

العديد من الأنشطة التجارية موسمية seasonal بطبيعتها، حيث تعتمد معظم الأعمال على وقت معين من المهرجان والعطلات. تستخدم كل شركة تقنيات ترويج المبيعات لزيادة الطلب على منتجاتها وخدماتها من أجل البقاءفي السوق لفترة أطول.في هذه المقالة، سأقوم بالتنبؤ بالمبيعات باستخدام التعلم الآلي من خلال تحليل البيانات التاريخية باستخدام تقنيات مثل التنبؤ بالسلاسل الزمنية Time Series Forecasting.

التنبؤ المبيعات مع التنبؤ بالسلاسل الزمنية

البيانات التي سأستخدمها هنا للتنبؤ بالمبيعات، هي بيانات مبيعات أسبوعية لتسعة متاجر وثلاثة منتجات.في نهاية هذه المقالة، سأتنبأ بالمبيعات لمدة 50 أسبوعًا قادمة، والآن للمضي قدمًافي التنبؤ بالسلسلة الزمنية، يمكنك تنزيل هذه البيانات التي سأستخدمها أدناه.

تحميل مجموعة البيانات

الآن، لنبدأ باستيراد المكتبات القياسية وقراءة مجموعة البيانات:

```
import plotly.express as px
from fbprophet import Prophet
from sklearn.metrics import mean_squared_error
from math import sqrt
from statsmodels.distributions.empirical_distribution import
ECDF
import datetime
import pandas as pd
import numpy as np
df = pd.read_csv('Sales_Product_Price_by_Store.csv')

df['Date'] = pd.to_datetime(df['Date'])
df['weekly_sales'] = df['Price'] * df['Weekly_Units_Sold']
df.head()
```

	Store	Product	Date	Is_Holiday	Base Price	Price	Weekly_Units_Sold	weekly_s
0	1	1	2010-02-05	False	9.99	7.99	245	1957.55
1	1	1	2010-02-12	True	9.99	7.99	453	3619.47
2	1	1	2010-02-19	False	9.99	7.99	409	3267.91
3	1	1	2010-02-26	False	9.99	7.99	191	1526.09
4	1	1	2010-03-05	False	9.99	9.99	145	1448.55

```
df.set_index('Date', inplace=True)
df['year'] = df.index.year
df['month'] = df.index.month
df['day'] = df.index.day
df['week_of_year'] = df.index.weekofyear
df.head()
```

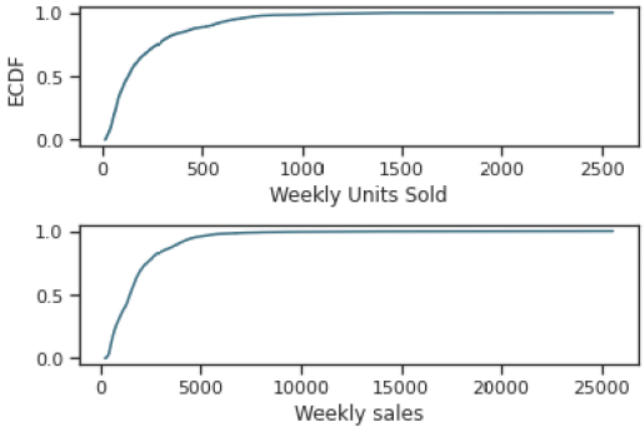
Date	Store	Product	Is_Holiday	Base Price	Price	Weekly_Units_Sold	weekly_s
2010-02-05	1	1	False	9.99	7.99	245	1957.55
2010-02-12	1	1	True	9.99	7.99	453	3619.47
2010-02-19	1	1	False	9.99	7.99	409	3267.91
2010-02-26	1	1	False	9.99	7.99	191	1526.09
2010-03-05	1	1	False	9.99	9.99	145	1448.55

تحليل البيانات الاستكشافية (EDA)

للحصول على بعض الأفكار حول المتغيرات المستمرةفي البيانات، سوف أرسم دالة التوزيع التجريبية (empirical distribution function (ECDF:

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style = "ticks")
c = '#386B7F'
figure, axes = plt.subplots(nrows=2, ncols=2)
figure.tight_layout(pad=2.0)
```

```
plt.subplot(211)
cdf = ECDF(df['Weekly_Units_Sold'])
plt.plot(cdf.x, cdf.y, label = "statmodels", color = c);
plt.xlabel('Weekly Units Sold'); plt.ylabel('ECDF');

plt.subplot(212)
cdf = ECDF(df['weekly_sales'])
plt.plot(cdf.x, cdf.y, label = "statmodels", color = c);
plt.xlabel('Weekly sales');
```



يوضح الشكل أعلاه بوضوح أنهفي أفضل أسبوع للمبيعات، تمكن المتجر من بيع 2500 وحدة، ولكن حوالي 80في المائة من الوقت، لم تتجاوز المبيعات الأسبوعية 500 وحدة.

لرؤية هذا بالأرقام، دعنا نلقي نظرة على إحصائيات بيانات المبيعات لدينا:

```
df.groupby('Store')['weekly_sales'].describe()
```

	count	mean	std	min	25%	50%	75%	max	
Store									
1	429.0	1789.414172	900.074226	769.65	1208.90	1659.17	1957.20	681(
2	429.0	2469.447413	1328.162884	1143.48	1579.21	2215.08	2756.55	911(
3	429.0	670.924009	366.816321	229.77	459.77	619.69	730.78	265(
4	429.0	3078.462145	1746.147872	1099.45	1818.18	2626.61	3837.51	137(
5	429.0	588.922984	242.628977	285.87	461.23	519.74	613.53	226(
6	429.0	2066.705082	1163.284768	890.19	1418.58	1758.40	2156.40	793(
7	429.0	955.115058	489.084883	389.61	649.35	857.61	1041.51	327(
8	429.0	1352.094056	811.326288	516.53	846.23	1275.87	1491.51	665(
10	429.0	4093.407249	3130.087191	1483.65	2462.88	3707.81	4510.47	255(

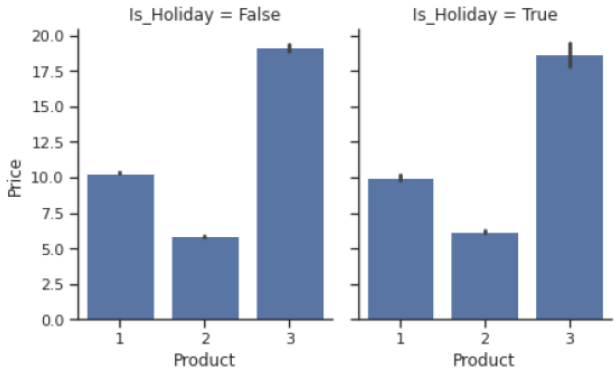
```
df.groupby('Store')['Weekly_Units_Sold'].sum()
```

```
Store
1      86699
2     121465
3      31689
4     158718
5      27300
6      97698
7      44027
8      65273
10    200924
Name: Weekly_Units_Sold, dtype: int64
```

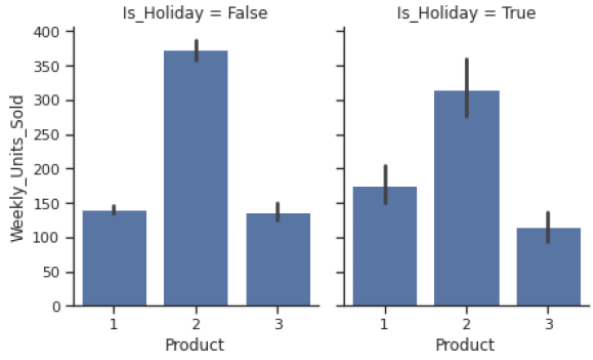
يوضح الشكل أعلاه بوضوح أنهفي أفضل أسبوع للمبيعات، تمكن المتجر من بيع 2500 وحدة، ولكن حوالي 80في المائة من الوقت، لم تتجاوز المبيعات الأسبوعية 500 وحدة.

لرؤية هذا بالأرقام، دعنا نلقي نظرة على إحصائيات بيانات المبيعات لدينا:

```
g = sns.FacetGrid(df, col="Is_Holiday", height=4, aspect=.8)
g.map(sns.barplot, "Product", "Price")
```



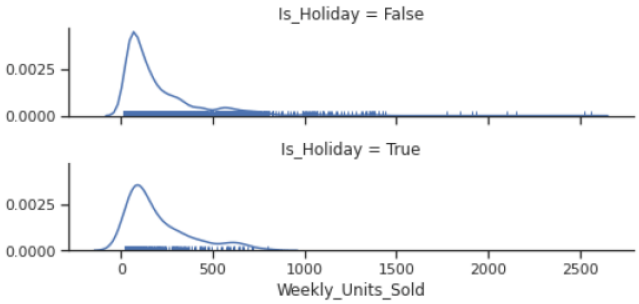

```
g = sns.FacetGrid(df, col="Is_Holiday", height=4, aspect=.8)
g.map(sns.barplot, "Product", "Weekly_Units_Sold")
```



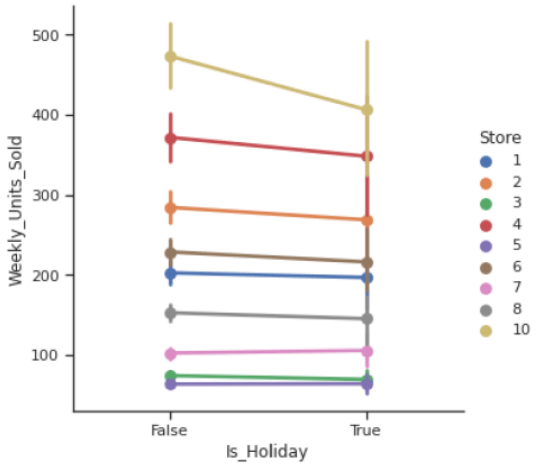
المنتج 2 هو أرخص منتج من بين هذه المنتجات الثلاثة، لذا فهو الأكثر مبيعًا. المنتج 3 هو أغلى منتج بين هؤلاء الثلاثة. لم يتغير سعر المنتج خلال العطلات.

نظرًا لأننا سجلنا مبيعات أيام العطلات، فسنقوم بتحليل ما إذا كانت العطلة قد ساهمت أيضًا في المبيعات.

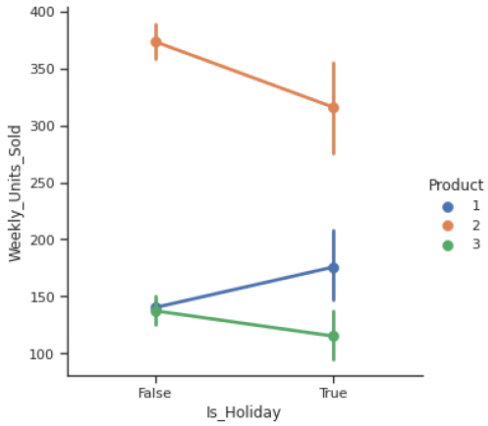
```
G = sns.FacetGrid(df, row="Is_Holiday",
                  height=1.7, aspect=4,)
g.map(sns.distplot, "Weekly_Units_Sold", hist=False, rug=True)
```




```
sns.factorplot(data= df,
               x= 'Is_Holiday',
               y= 'Weekly_Units_Sold',
               hue= 'Store')
```



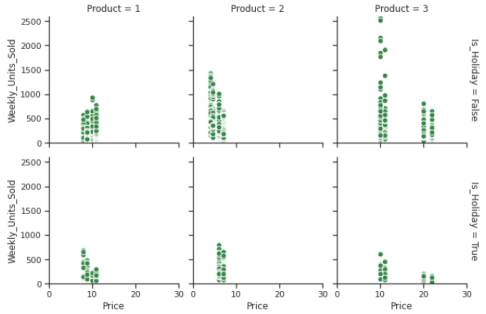

```
sns.factorplot(data= df,
               x= 'Is_Holiday',
               y= 'Weekly_Units_Sold',
               hue= 'Product')
```



من الأرقام المذكورة أعلاه يمكننا أن نرى أن العطلات ليس لها تأثير إيجابي على العمل. بالنسبة لمعظم المتاجر، تكون مبيعات الوحدات الأسبوعيةفي أيام العطلات مماثلة للأيام العادية، بينما يواجه المتجر 10 أيضًا انخفاضًافي المبيعات خلال العطلات.

شهدت الوحدات الأسبوعية المباعة للمنتج 1 زيادة طفيفة خلال العطلات، بينما انخفض المنتج 2 والمنتج 3 خلال العطلات.

```
G = sns.FacetGrid(df, col="Product", row="Is_Holiday",
margin_titles=True, height=3)
g.map(plt.scatter, "Price", "Weekly_Units_Sold",
color="#338844", edgecolor="white", s=50, lw=1)
g.set(xlim=(0, 30), ylim=(0, 2600));
```

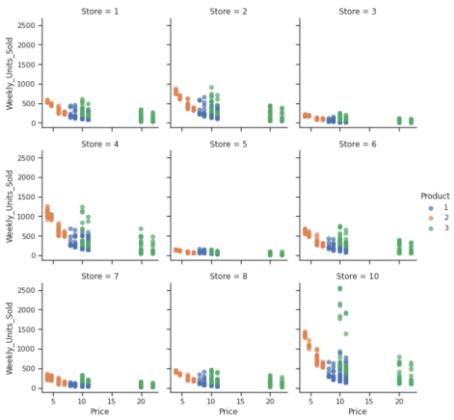


كل منتج له أكثر من سعر، سواءفي أيام العطلات أوفي الأيام العادية. أحد الأسعار سعر عادي regular price والآخر سعر ترويجي promotional price. ومع ذلك، فإن الفجوة السعرية للمنتج 3 ضخمة، فقد تم تخفيضها إلى ما يقرب من 50٪ أثناء العروض الترويجية.

حقق المنتج 3 أكبر قدر من المبيعات خلال الأيام العادية.

```
G = sns.FacetGrid(df, col="Store", hue="Product",
margin_titles=True, col_wrap=3)
```

```
g.map(plt.scatter, 'Price', 'Weekly_Units_Sold', alpha=.7)
g.add_legend()
```



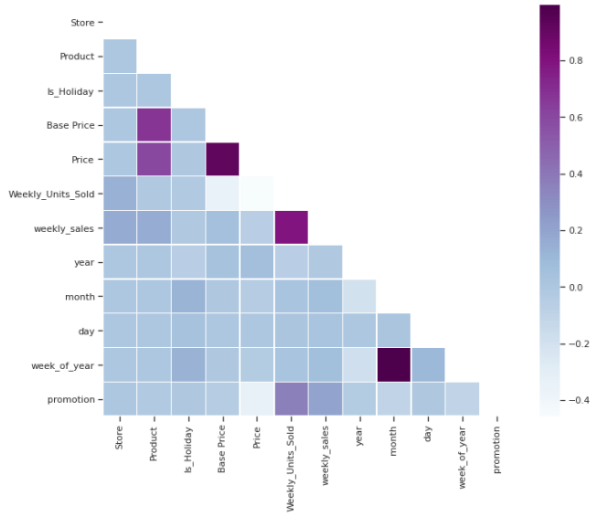
جميع المتاجر لديها نمط ترويج الأسعار المماثل، لسبب ما، يبيع المتجر 10 أكثر خلال العروض الترويجية. جميع المنتجات لديها السعر العادي وسعر الترويج. يتمتع المنتج 3 بأعلى خصم ويتم بيعه بأكبر قدر خلال فترة العروض الترويجية.

```
Df.groupby(['Product', 'promotion'])['Price',
'Weekly_Units_Sold'].mean()
```

Product	promotion		
1	0	10.653866	131.637722
	1	8.523333	199.171296
2	0	6.294348	317.388406
	1	4.201429	581.099206
3	0	20.700744	87.748393
	1	10.409091	400.484848

الآن، دعنا ننشئ خريطة حرارية heatmap لاستكمال جميع ملاحظاتنا:

```
corr_all = df.corr()

# Generate a mask for the upper triangle
mask = np.zeros_like(corr_all, dtype = np.bool)
mask[np.triu_indices_from(mask)] = True

# Set up the matplotlib figure
f, ax = plt.subplots(figsize = (11, 9))

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr_all, mask = mask,
            square = True, linewidths = .5, ax = ax, cmap =
"BuPu")
plt.show();
```



لدينا علاقة إيجابية قوية بين السعر والسعر الأساسي، والوحدات المباعة الأسبوعية والمبيعات الأسبوعية، والسعر الأساسي والمنتج، والسعر والمنتج. يمكننا أيضًا ملاحظة وجود ارتباط إيجابي بين شهر وأسبوع السنة.

ملاحظات من تحليل البيانات الاستكشافية:

- المتجر الأكثر مبيعًا وازدحامًا هو المتجر 10، وأقل المتاجر ازدحامًا هو المتجر 5.
- من حيث عدد الوحدات المباعة، المنتج الأكثر مبيعًا هو المنتج 2. من حيث دولارات المبيعات، يسجل المنتج 3 أعلى مبيعات خلال الأيام العادية.
- لا تقوم المتاجر بالضرورة بإجراء عروض ترويجية للمنتج خلال العطلات. لا يبدو أن للعطلات تأثير على أداء المتاجر.
- يُباع المنتج 1 أكثر قليلاً خلال العطلات، ومع ذلك، فإن المنتج 2 والمنتج 3 يبيعان أقل في أيام العطلات.
- يبدو أن المنتج 2 هو أرخص منتج، والمنتج 3 هو أغلى منتج.
- تتمتع معظم المتاجر بنوع من الموسمية وتحقق أعلى مبيعات في شهر يوليو تقريبًا.
- تم بيع المنتج 1 أكثر بقليل في فبراير مقارنة بالأشهر الأخرى، وكان المنتج 2 أكثر بيعًا في شهري أبريل ويوليو، وكان المنتج 3 مبيعًا في شهر يوليو تقريبًا.
- بشكل عام، تم بيع المنتج 2 بأكبر قدر في المتجر 10، ولكن في يوليو، حقق المنتج 3 أعلى مبيعات في هذا المتجر.

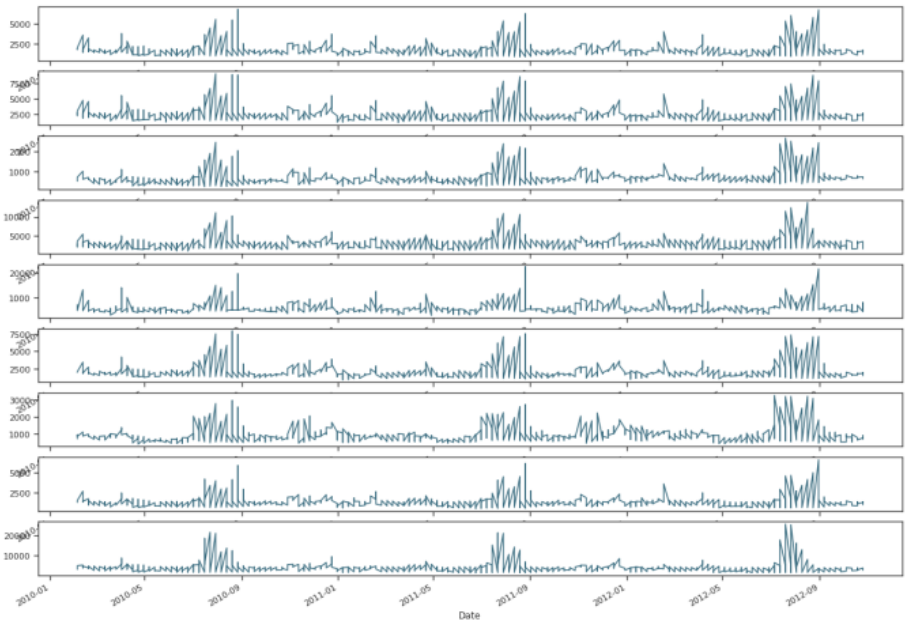
- كل منتج له سعره المعتاد وسعره الترويجي. لا توجد فجوة كبيرة بين السعر العادي والسعر الترويجي للمنتج 1 والمنتج 2، ومع ذلك، يمكن خفض السعر الترويجي للمنتج 3 إلى 50٪ من سعره الأصلي. على الرغم من أن كل متجر يقوم بتخفيض هذا النوع من الأسعار للمنتج 3، إلا أن المتجر 10 هو الذي حقق أعلى مبيعات خلال خفض الأسعار.

- ليس من غير المألوف أن تبيع أثناء الترويج أكثر من الأيام العادية. جعل المتجر العاشر المنتج 3 المنتج الأكثر مبيعًافي شهر يوليو تقريبًا.

التنبؤ بالسلاسل الزمنية والتنبؤ بالمبيعات

الآن دعنا ننتقل إلى جزء التنبؤ بالسلسلة الزمنية من هذه المقالة، هنا سوف نتوقع المبيعات، وفقًا لملاحظاتنا أعلاه لتحليل البيانات الاستكشافية.

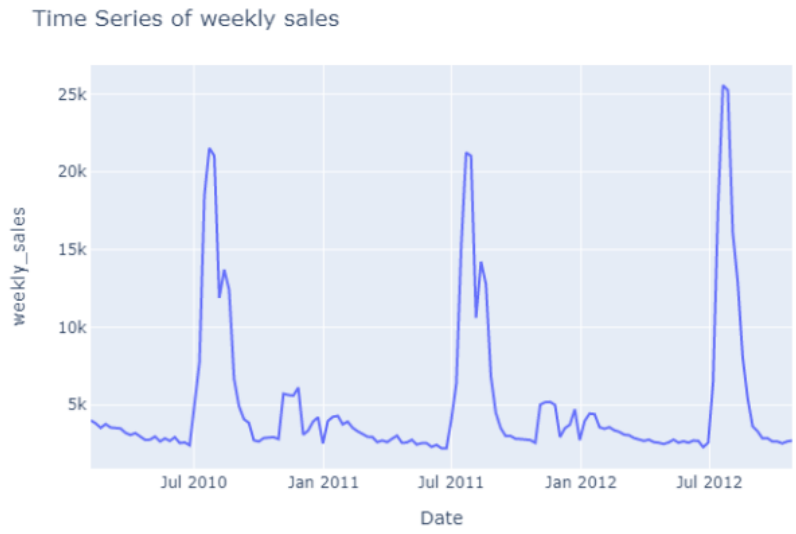
```
# store types
sales_1 = df[df.Store == 1]['weekly_sales']
sales_2 = df[df.Store == 2]['weekly_sales']
sales_3 = df[df.Store == 3]['weekly_sales']
sales_4 = df[df.Store == 4]['weekly_sales']
sales_5 = df[df.Store == 5]['weekly_sales']
sales_6 = df[df.Store == 6]['weekly_sales']
sales_7 = df[df.Store == 7]['weekly_sales']
sales_8 = df[df.Store == 8]['weekly_sales']
sales_10 = df[df.Store == 10]['weekly_sales']

f, (ax1, ax2, ax3, ax4, ax5, ax6, ax7, ax8, ax9) =
plt.subplots(9, figsize = (20, 15))

# store types
sales_1.plot(color = c, ax = ax1)
sales_2.plot(color = c, ax = ax2)
sales_3.plot(color = c, ax = ax3)
sales_4.plot(color = c, ax = ax4)
sales_5.plot(color = c, ax = ax5)
sales_6.plot(color = c, ax = ax6)
sales_7.plot(color = c, ax = ax7)
sales_8.plot(color = c, ax = ax8)
sales_10.plot(color = c, ax = ax9)
```



التنبؤ بالسلاسل الزمنية

السلاسل الزمنية للمبيعات الأسبوعية:

```
store_10_pro_3 = df[(df.Store == 10) &amp;amp; (df.Product ==
3)].loc[:, ['Base Price', 'Price', 'Weekly_Units_Sold',
'weekly_sales']]
store_10_pro_3.reset_index(level=0, inplace=True)
fig = px.line(store_10_pro_3, x='Date', y='weekly_sales')
fig.update_layout(title_text='Time Series of weekly sales')
fig.show()
```



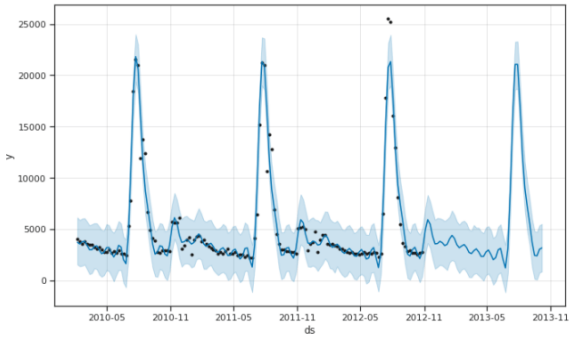
الموسمية للمنتج 2في المتجر 10 واضحة. تبلغ المبيعات ذروتها دائمًا بين شهري يوليو وسبتمبر خلال العطلة المدرسية. فيما يلي نطبق نموذج prophet، ونتوقع المبيعات الأسبوعية للأسبوع الخمسين القادمة.

```
Model = Prophet(interval_width = 0.95)
model.fit(store_10_pro_3)

future_dates = model.make_future_dataframe(periods = 50,
freq='W')

future_dates.tail(7)
```

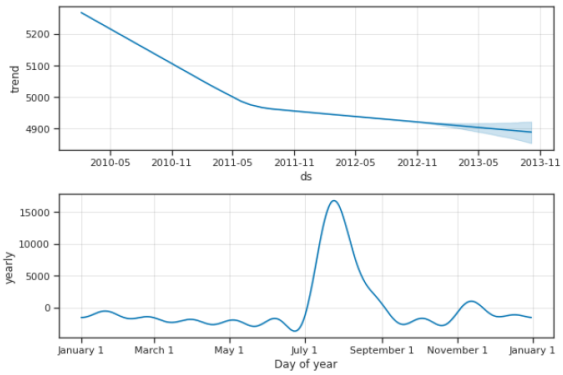
```
forecast = model.predict(future_dates)

# preditions for last week
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail(7)
```

	ds	yhat	yhat_lower	yhat_upper
186	2013-08-25	7160.453669	4742.937710	9559.615673
187	2013-09-01	5542.434739	3249.762712	7887.321785
188	2013-09-08	3702.168377	1355.902566	5824.555193
189	2013-09-15	2427.279755	189.552142	4693.158976
190	2013-09-22	2386.972428	7.973471	4673.053027
191	2013-09-29	3020.451351	759.252236	5227.695107
192	2013-10-06	3157.655085	756.079499	5603.923897

```
model.plot(forecast)
```




```
model.plot_components(forecast)
```




```
metric_df =
forecast.set_index('ds')[['yhat']].join(store_10_pro_3.set_ind
ex('ds').y).reset_index()
metric_df.dropna(inplace=True)
error = mean_squared_error(metric_df.y, metric_df.yhat)
print('The RMSE is {}'. Format(sqrt(error)))
```

```
The RMSE is 1190.0962582193933
```

آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بالسلسلة الزمنية على التنبؤ بالمبيعات.

المصدر:

https://thecleverprogrammer.com/2020/07/11/time-series-forecasting/

10) التنبؤ بالطقس باستخدام التعلم الآلي Predict Weather with Machine Learning

في هذه المقالة، سأقوم بتدريب نموذج للتنبؤ بالطقس باستخدام التعلم الآلي. سنتصرف كما لو أننا لا نستطيع الوصول إلى توقعات الطقس. لدينا إمكانية الوصول إلى قرن من المتوسطات التاريخية لدرجات الحرارة العالمية، بمافي ذلك درجات الحرارة العالمية القصوى، ودرجات الحرارة الدنيا العالمية، ودرجات حرارة الأرض والمحيطات العالمية. بعد كل هذا، نعلم أن هذه مشكلة تعلم آلة انحدار خاضعة للإشراف.

مجموعة بيانات الطقس للتنبؤ بالطقس

بادئ ذي بدء، نحتاج إلى بعض البيانات، تم إنشاء البيانات التي أستخدمها للتنبؤ بالطقس باستخدام التعلم الآلي من إحدى الجامعات البحثية المرموقةفي العالم، وسنفترض أن البيانات الموجودةفي مجموعة البيانات صحيحة. يمكنك بسهولة تنزيل هذه البيانات من هنا.

الآن، دعنا نبدأفي قراءة مجموعة البيانات:

```
import pandas as pd
global_temp = pd.read_csv("GlobalTemperatures.csv")
print(global_temp.shape)
print(global_temp.columns)
print(global_temp.info())
print(global_temp.isnull().sum())
```

تحضير البيانات

لسوء الحظ، لم نصل تمامًا إلى المرحلة التي يمكننا فيها فقط إدخال البيانات الأوليةفي نموذج وجعله يرسل ردًا. سنحتاج إلى إجراء بعض التعديلات الطفيفة لوضع بياناتنافي نموذج التعلم الآلي.

ستعتمد الخطوات الدقيقةفي إعداد البيانات على النموذج المستخدم والبيانات التي تم جمعها، ولكن ستكون هناك حاجة إلى قدر من معالجة البيانات. أولاً، سأُنشئ دالة تسمى wrangle () ساستدعي فيها على dataframe:

```
#Data Preparation
def wrangle(df):
    df = df.copy()
    df = df.drop(columns=["LandAverageTemperatureUncertainty",
"LandMaxTemperatureUncertainty",
                          "LandMinTemperatureUncertainty",
"LandAndOceanAverageTemperatureUncertainty"], axis=1)
```

نريد عمل نسخة من إطار البيانات حتى لا نتلف الأصل. بعد ذلك، سنقوم بإزالة الأعمدة ذات العلاقة الأساسية العالية high cardinality.

تشير العلاقة الأساسية العالية إلى الأعمدة التي تكون قيمها نادرة جدًا أو فريدة. نظرًا لتكرار البيانات الأساسية العاليةفي معظم مجموعات بيانات السلاسل الزمنية، سنحل هذه المشكلة مباشرةً عن طريق إزالة أعمدة عدد العناصر العالية هذه تمامًا من مجموعة البيانات الخاصة بنا حتى لا يتم الخلط بين نموذجنافي المستقبل.

الآن، سأقوم بإنشاء دالة لتحويل درجة الحرارة، ولتحويل الأعمدة إلى كائن DateTime:

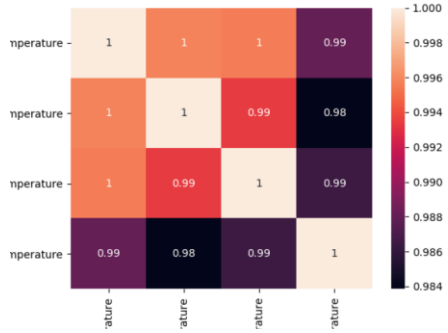
```
def converttemp(x):
        x = (x * 1.8) + 32
        return float(x)
    df["LandAverageTemperature"] =
df["LandAverageTemperature"].apply(converttemp)
    df["LandMaxTemperature"] = df["LandMaxTemperature"].apply(converttemp)
    df["LandMinTemperature"] = df["LandMinTemperature"].apply(converttemp)
    df["LandAndOceanAverageTemperature"] =
df["LandAndOceanAverageTemperature"].apply(converttemp)
    df["dt"] = pd.to_datetime(df["dt"])
    df["Month"] = df["dt"].dt.month
    df["Year"] = df["dt"].dt.year
    df = df.drop("dt", axis=1)
    df = df.drop("Month", axis=1)
    df = df[df.Year &gt;= 1850]
    df = df.set_index(["Year"])
    df = df.dropna()
    return df
global_temp = wrangle(global_temp)
print(global_temp.head())
```

بعد استدعاء دالة wrangle الخاصة بنا إلى global_temp dataframe، يمكننا الآن رؤية نسخة منظَّفة جديدة من إطار البيانات global_temp الخاص بنا بدون قيم مفقودة.

رسم البيانات

الآن، قبل المضي قدمًافي تدريب نموذج للتنبؤ بالطقس باستخدام التعلم الآلي، دعنا نرسم هذه البيانات للعثور على الارتباطات بين البيانات:

```
import seaborn as sns
import matplotlib.pyplot as plt
corrMatrix = global_temp.corr()
sns.heatmap(corrMatrix, annot=True)
plt.show()
```



كما يمكننا أن نرى، وكما قد خمّن بعضكم على الأرجح، فإن الأعمدة التي اخترناها للاستمرار في المضي قدمًا مرتبطة ارتباطًا وثيقًا ببعضها البعض.

فصل هدفنا للتنبؤ بالطقس

نحتاج الآن إلى فصل البيانات إلى ميزات وأهداف. الهدف، المسمى أيضًا Y، هو القيمة التي نريد توقعها،في هذه الحالة، المتوسط الفعلي لدرجة حرارة الأرض والمحيطات والميزات هي جميع الأعمدة التي يستخدمها النموذج لإجراء التنبؤ:

```
target = "LandAndOceanAverageTemperature"
y = global_temp[target]
x = global_temp[["LandAverageTemperature", "LandMaxTemperature",
"LandMinTemperature"]]
```

التقسيم الى بيانات تدريب واختبار

الآن، لإنشاء نموذج للتنبؤ بالطقس باستخدام التعلم الآلي، نحتاج إلى تقسيم البيانات باستخدام طريقة train_test_split المقدمة من scikit-Learn:

```
from sklearn.model_selection import train_test_split
xtrain, xval, ytrain, yval = train_test_split(x, y, test_size=0.25,
random_state=42)
print(xtrain.shape)
print(xval.shape)
print(ytrain.shape)
print(yval.shape)
```

```
(1494, 3)
(498, 3)
(1494,)
(498,)
```

خط أساس متوسط الخطأ المطلق

قبل أن نتمكن من إجراء وتقييم أي تنبؤات على نموذج التعلم الآلي الخاص بنا للتنبؤ بالطقس، نحتاج إلى إنشاء خط أساس Baseline، وهو مقياس عاقل نأملفي التغلب عليه باستخدام نموذجنا. إذا لم يتمكن نموذجنا من التحسن من خط الأساس، فسوف يفشل ويجب أن نجرب نموذجًا مختلفًا أو نعترف بأن التعلم الآلي غير مناسب لمشكلتنا:

```
from sklearn.metrics import mean_squared_error
ypred = [ytrain.mean()] * len(ytrain)
print("Baseline MAE: ", round(mean_squared_error(ytrain, ypred), 5))
```

تدريب النموذج للتنبؤ بالطقس

الآن للتنبؤ بالطقس باستخدام التعلم الآلي، سأقوم بتدريب خوارزمية Random Forest قادرة على أداء مهام التصنيف وكذلك الانحدار:

```
from sklearn.feature_selection import SelectKBest
from sklearn.ensemble import RandomForestRegressor
forest = make_pipeline(
```

```
    SelectKBest(k="all"),
    StandardScaler(),
    RandomForestRegressor(
        n_estimators=100,
        max_depth=50,
        random_state=77,
        n_jobs=-1
    )
)
forest.fit(xtrain, ytrain)
```

تقييم نموذج التعلم الآلي للتنبؤ بالطقس

لوضع تنبؤاتنافي منظورها الصحيح، يمكننا حساب precision باستخدام متوسط النسبة المئوية للخطأ مطروحًا من 100٪:

```
import numpy as np
errors = abs(ypred - yval)
mape = 100 * (errors/ytrain)
accuracy = 100 - np.mean(mape)
print("Random Forest Model: ", round(accuracy, 2), "%")
```

```
Random Forest Model: 99.52 %
```

الملخص

لقد تعلم نموذجنا التنبؤ بالظروف الجوية باستخدام التعلم الآلي للعام المقبل بدقة تصل إلى 99٪. آمل أن تكون قد أحببت هذه المقالة حول كيفية بناء نموذج للتنبؤ بالطقس باستخدام التعلم الآلي.

المصدر:

https://thecleverprogrammer.com/2020/08/30/predict-weather-with-/machine-learning

11) السلاسل الزمنية مع LSTM في التعلم الآلي Time Series with LSTM in Machine Learning

يمكن أن تكون الشبكات العصبية Neural networks مفهومًا صعب الفهم. أعتقد أن هذا يرجع أساسًا إلى إمكانية استخدامها للعديد من الأشياء المختلفة مثل التصنيف classification أو التحديد identification أو الانحدار regression فقط.في هذه المقالة، سوف أطلعك على كيفية إعداد طريقة بسيطة للتنبؤ بالسلاسل الزمنية باستخدام نموذج LSTM.

قبل البدءفي جزء البرمجة للتنبؤ بالسلسلة الزمنية باستخدام LSTM أولاً، دعنا ننتقل إلى بعض المفاهيم الرئيسية المتضمنة لجميع المبتدئين الذين يقرؤون هذه المقالة.

ما هو التنبؤ بالسلاسل الزمنية؟

التنبؤ بالسلسلة الزمنية Time series forecasting هو أسلوب للتنبؤ بالأحداث من خلال تسلسل زمني. تستخدم هذه التقنيةفي العديد من مجالات الدراسة، من الجيولوجيا إلى السلوك إلى الاقتصاد. تتنبأ التقنيات بالأحداث المستقبلية من خلال تحليل الاتجاهات من الماضي، على افتراض أن الاتجاهات المستقبلية ستظل مماثلة للاتجاهات التاريخية.

ما هو LSTM؟

LSTM تعني الذاكرة قصيرة المدى طويلة المدى Short-Term Long-Term Memory. إنه نموذج أو معمارية توسع ذاكرة الشبكات العصبية المتكررة recurrent neural networks. عادةً ما تحتوي الشبكات العصبية المتكررة على "ذاكرة قصيرة المدى" من حيث أنها تستخدم معلومات سابقة ثابتة لاستخدامهافي الشبكة العصبية الحالية. بشكل أساسي، يتم استخدام المعلومات السابقةفي المهمة الحالية. هذا يعني أنه ليس لدينا قائمة بجميع المعلومات السابقة المتاحة للعقدة العصبية.

السلاسل الزمنية للتنبؤ مع LSTM

آمل أن تكون قد فهمت ما يعنيه التنبؤ بالسلاسل الزمنية وما هي نماذج LSTM. الآن سوف أتجه نحو إنشاء نموذج التعلم الآلي للتنبؤ بالسلسلة الزمنية باستخدام LSTMفي التعلم الآلي.

لكي تتنبأ هذه المهمة بالسلسلة الزمنية باستخدام LSTM، سأبدأ باستيراد جميع الحزم الضرورية التي نحتاجها:

```
import numpy
import matplotlib.pyplot as plt
import pandas
import math
from keras.models import Sequential
from keras.layers import Dense
```

```
from keras.layers import LSTM
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
# fix random seed for reproducibility
numpy.random.seed(7)
```

الآن دعنا نحمّل البيانات ونجهز البيانات حتى نتمكن من استخدامها على نموذج LSTM، يمكنك تنزيل مجموعة البيانات التي أستخدمهافي هذه المهمة من هنا:

```
# load the dataset
dataframe = pandas.read_csv('airline-passengers.csv',
usecols=[1], engine='python')
dataset = dataframe.values
dataset = dataset.astype('float32')
# normalize the dataset
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset)
```

الآن، سأقسم البيانات إلى مجموعات تدريب training sets ومجموعات اختبار test sets:

```
# split into train and test sets
train_size = int(len(dataset) * 0.67)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:],
dataset[train_size:len(dataset),:]
print(len(train), len(test))
```

```
96 48
```

السلاسل الزمنية مع LSTM

الآن قبل تدريب البيانات على نموذج LSTM، نحتاج إلى إعداد البيانات حتى نتمكن من ملاءمتها (fit)في النموذج، وسأحدد هذه المهمة دالة مساعدة:

```
# convert an array of values into a dataset matrix
def create_dataset(dataset, look_back=1):
	dataX, dataY = [], []
	for i in range(len(dataset)-look_back-1):
		a = dataset[i:(i+look_back), 0]
		dataX.append(a)
		dataY.append(dataset[i + look_back, 0])
	return numpy.array(dataX), numpy.array(dataY)
```

الآن، نحتاج إلى إعادة تشكيل (reshape) البيانات قبل تطبيقهافي نموذج LSTM:

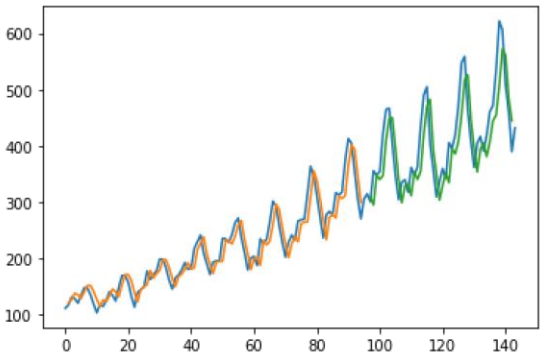
```
# reshape into X=t and Y=t+1
look_back = 1
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)
# reshape input to be [samples, time steps, features]
trainX = numpy.reshape(trainX, (trainX.shape[0], 1,
trainX.shape[1]))
testX = numpy.reshape(testX, (testX.shape[0], 1,
testX.shape[1]))
```

الآن بعد اكتمال جميع المهام المتعلقة بإعداد البيانات لتلائم نموذج LSTM، حان الوقت لملاءمة البيانات الموجودةفي النموذج ودعنا ندرب النموذج:

```
# create and fit the LSTM network
model = Sequential()
model.add(LSTM(4, input_shape=(1, look_back)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)
```

الآن، دعنا نتنبأ ونرسم اتجاهات السلاسل الزمنية باستخدام حزمة matplotlibفي بايثون:

```
# make predictions
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)
# invert predictions
trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform([trainY])
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform([testY])
# calculate root mean squared error
trainScore = math.sqrt(mean_squared_error(trainY[0],
trainPredict[:,0]))
testScore = math.sqrt(mean_squared_error(testY[0],
testPredict[:,0]))

# shift train predictions for plotting
trainPredictPlot = numpy.empty_like(dataset)
trainPredictPlot[:, :] = numpy.nan
trainPredictPlot[look_back:len(trainPredict)+look_back, :] =
trainPredict
# shift test predictions for plotting
testPredictPlot = numpy.empty_like(dataset)
testPredictPlot[:, :] = numpy.nan
testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)
-1, :] = testPredict
# plot baseline and predictions
plt.plot(scaler.inverse_transform(dataset))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.show()
```



آمل أن تكون قد أحببت هذه المقالة حول التنبؤ بالسلسلة الزمنية باستخدام نموذج LSTM.

المصدر:

https://thecleverprogrammer.com/2020/08/29/time-series-with-lstm-/in-machine-learning

12) التنبؤ بالمواليد اليومية باستخدام التعلم الآلي Daily Births Forecasting with Machine Learning

في هذه المقالة، سأستخدم الخوارزمية التي يوفرها Facebook، والمعروفة باسم Facebook Prophet Model. سأستخدم نموذج Facebook Prophet للتنبؤ بالمواليد اليومية باستخدام التعلم الآلي. البيانات التي سأستخدمها هنا هي مجموعة بيانات مشهورة جدًا بين ممارس التعلم الآلي المعروف باسم الولادات اليومية للإناثفي كاليفورنيا.

قبل البدءفي مهمة التنبؤ بالمواليد اليومية باستخدام التعلم الآلي، اسمحوا لي أن أقدم لكم نموذج Facebook Prophet، حيث سأستخدم نموذج Facebook Prophetفي هذه المقالة.

ما هو Facebook Prophet؟

Facebook Prophet عبارة عن خوارزمية تم تطويرها بواسطة فريق Core Data Science في Facebook. يتم استخدامهفي تطبيقات التنبؤ بالسلاسل الزمنية. يتم استخدامه كثيرًا عندما يكون هناك احتمال حدوث تأثيرات موسمية.في هذه المقالة، سوف آخذك عبر تطبيق نموذج Facebook Prophet للتنبؤ بالمواليد اليومية باستخدام التعلم الآلي.

التنبؤ بالمواليد اليومية

دعنا نبدأ بمهمة التنبؤ بالمواليد اليومية باستخدام التعلم الآلي باستخدام نموذج Facebook Prophet. سأبدأ بهذه المهمة عن طريق استيراد جميع الحزم الضرورية التي نحتاجها لهذه المهمة:

```
import pandas as pd
import numpy as np
import fbprophet
from fbprophet.plot import add_changepoints_to_plot
import warnings
import matplotlib.pyplot as plt
```

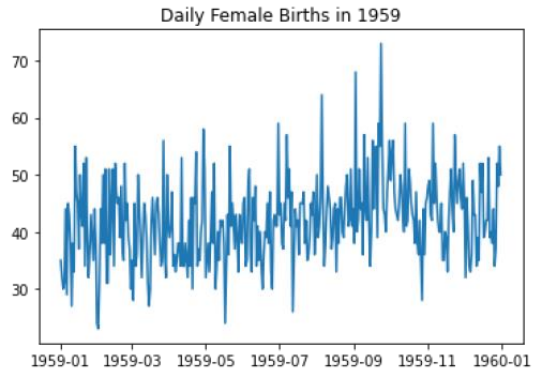
الآن، نظرًا لأنني قمت باستيراد جميع الحزم الضرورية، فسوف أمضي قدمًا من خلال قراءة مجموعة البيانات التي نحتاجها للتنبؤ بالمواليد اليومية:

```
df = pd.read_csv("daily-total-female-births.csv",
parse_dates=['Date'], date_parser=pd.to_datetime)
df.columns = ['ds', 'y']
df.head()
```

	ds	y
0	1959-01-01	35
1	1959-01-02	32
2	1959-01-03	30
3	1959-01-04	31
4	1959-01-05	44

لقد استخدمت "ds" و "y" كأسماء للأعمدة لأنها الطريقة المنسقة مسبقًا التي نحتاجها لملاءمة بياناتنافي نموذج Facebook Prophet. لذلك آمل ألا تشعر بالارتباك حيال ذلك. الآن، قبل استخدام خوارزمية FB prophet على بياناتنا، دعنا نرسم البيانات لإلقاء نظرة سريعة على ما نعمل به:

```
plt.plot(df['ds'], df['y']);
plt.title('Daily Female Births in 1959')
```

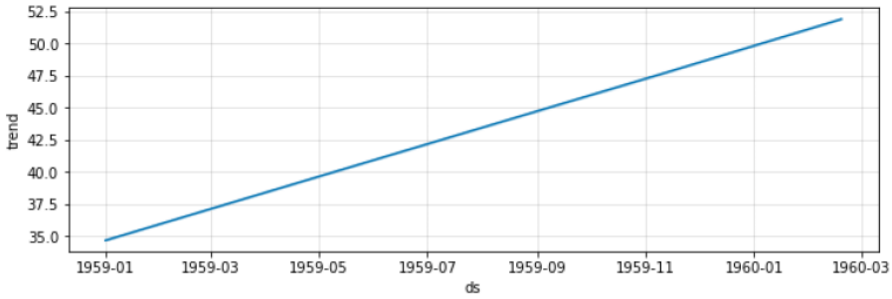


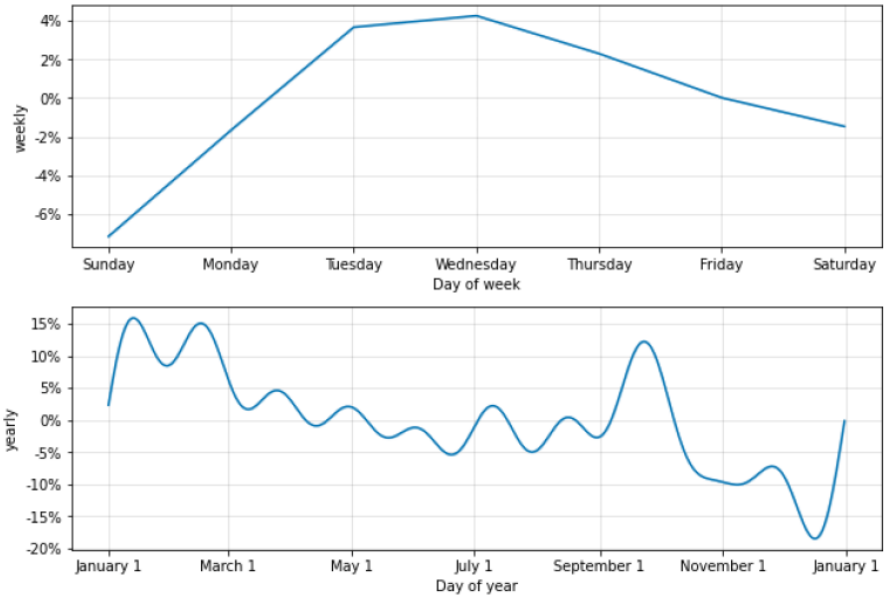
الآن، سوف أقوم بإنشاء مثيل Prophet لتطبيق التأثيرات الموسمية لمهمة التنبؤ بالمواليد اليومية باستخدام التعلم الآلي:

```
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    m = fbprophet.Prophet(yearly_seasonality=True,
daily_seasonality=False,
                          changepoint_range=0.9,
                          changepoint_prior_scale=0.5,
                          seasonality_mode='multiplicative')
    m.fit(df)
future = m.make_future_dataframe(periods=50, freq='d')
forecast = m.predict(future)
```

الآن، دعونا نرسم التأثيرات الموسمية التي حصلنا عليها بعد تطبيق النموذج:

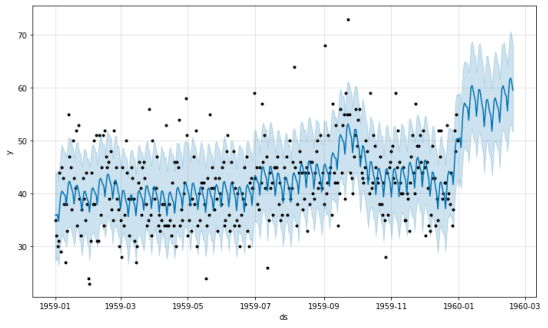
```
m.plot_components(forecast)
```





الآن، دعونا نرسم التنبؤات التي قدمها نموذج Facebook prophet للتنبؤ بالمواليد اليومية:

```
m.plot(forecast)
```



أتمنى أن تكون قد أحببت هذا المقالفي التنبؤ بالمواليد اليومية مع استخدام التعلم الآلي باستخدام نموذج Facebook Prophet.

المصدر:

https://thecleverprogrammer.com/2020/08/27/daily-births-forecasting-/with-machine-learning

13) التنبؤ بأسعار الأسهم مع نموذج Facebook Prophet Stock Price Prediction with Facebook Prophet Model

التنبؤ بسعر السهم Stock Price Prediction يعني تحديد القيمة المستقبلية للأسهم أو الأدوات المالية الأخرى للمؤسسة. إذا أتقنت فن التنبؤ بأسعار الأسهم، يمكنك كسب الكثير من خلال الاستثمار والبيع في الوقت المناسب، ويمكنك أيضًا أن تكسب من خلال توجيه الأشخاص الآخرين الذين يرغبون في استكشاف التداول.

نموذج Facebook Prophet

Facebook Prophet عبارة عن خوارزمية تم تطويرها بواسطة فريق Core Data Science في Facebook. يتم استخدامه في تطبيقات التنبؤ بالسلاسل الزمنية. يتم استخدامه كثيرًا عندما يكون هناك احتمال حدوث تأثيرات موسمية. يستخدم التنبؤ بالسلاسل الزمنية بشكل كبير في التنبؤ بسعر السهم. في هذا المقال سوف آخذكم من خلال تطبيق Facebook Prophet النموذجي الخاص بتنبؤ أسعار أسهم Google.

التنبؤ بأسعار الأسهم باستخدام نموذج Facebook Prophet

سأستخدم أحدث مجموعة بيانات قمت بتنزيلها للتو من موقع yahoo finance. إذا كنت ترغب في ممارسة هذه المهمة على نفس مجموعة البيانات التي أخذتها، فيمكنك تنزيلها من هنا.

إذا كنت ترغب في ممارسة هذا على أحدث مجموعة بيانات عندما تقرأ هذا المقال، فيمكنك تنزيل أحدثها من yahoo finance. إذا وجدت أي مشكلة في تنزيل أحدث مجموعة بيانات، يمكنك ذكرها في قسم التعليقات أدناه؛ سوف أساعدك في ذلك.

للتنبؤ بأسعار الأسهم باستخدام نموذج Facebook Prophet، يجب عليك تثبيت حزمة باسم fbprophet، والتي يمكن تثبيتها بسهولة باستخدام الأمر `pip install fbprophet`. أتمنى أن تكون قد قمت بتثبيت هذه الحزمة والآن دعنا ننتقل أكثر من خلال استيراد الحزم الضرورية التي نحتاجها لهذه المهمة:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pandas_datareader as web
import warnings
!pip install fbprophet
import fbprophet
```

الآن دعنا نقرأ ونلقي نظرة على البيانات التي نعمل معها:

```
from google.colab import files
uploaded = files.upload()
data = pd.read_csv("GOOG.csv")
```

```
data.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2019-08-09	1197.989990	1203.880005	1183.603027	1188.010010	1188.010010	1065700
1	2019-08-12	1179.209961	1184.959961	1167.671997	1174.709961	1174.709961	1003000
2	2019-08-13	1171.459961	1204.780029	1171.459961	1197.270020	1197.270020	1294400
3	2019-08-14	1176.310059	1182.300049	1160.540039	1164.290039	1164.290039	1578700
4	2019-08-15	1163.500000	1175.839966	1162.109985	1167.260010	1167.260010	1218700

قبل المضي قدمًا، دعنا نرسم البيانات حتى نتمكن من الحصول على رؤى أفضل حول البيانات التي سنعمل عليها:

```
plt.style.use("fivethirtyeight")
plt.figure(figsize=(16,8))
plt.title("Google Closing Stock Price")
plt.plot(data["Close"])
plt.xlabel("Date", fontsize=18)
plt.ylabel("Close Price USD ($)", fontsize=18)
plt.show()
```

هناك ميزتان فقط مطلوبتان من مجموعة البيانات وهما التاريخ Date وأسعار الإغلاق Close Prices. لذلك دعونا نجهز البيانات لنموذجنا:

```
data = data[["Date","Close"]]
data = data.rename(columns = {"Date":"ds","Close":"y"})
data.head()
```

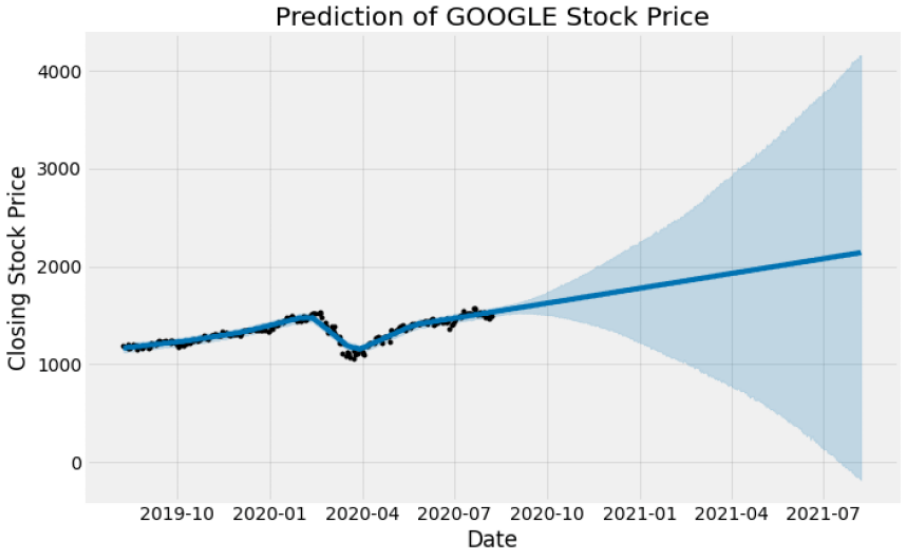
	ds	y
0	2019-08-09	1188.010010
1	2019-08-12	1174.709961
2	2019-08-13	1197.270020
3	2019-08-14	1164.290039
4	2019-08-15	1167.260010

الآن دعونا نلائم البيانات مع نموذج Facebook Prophet للتنبؤ بأسعار أسهم Google:

```
from fbprophet import Prophet
m = Prophet(daily_seasonality=True)
m.fit(data)
```

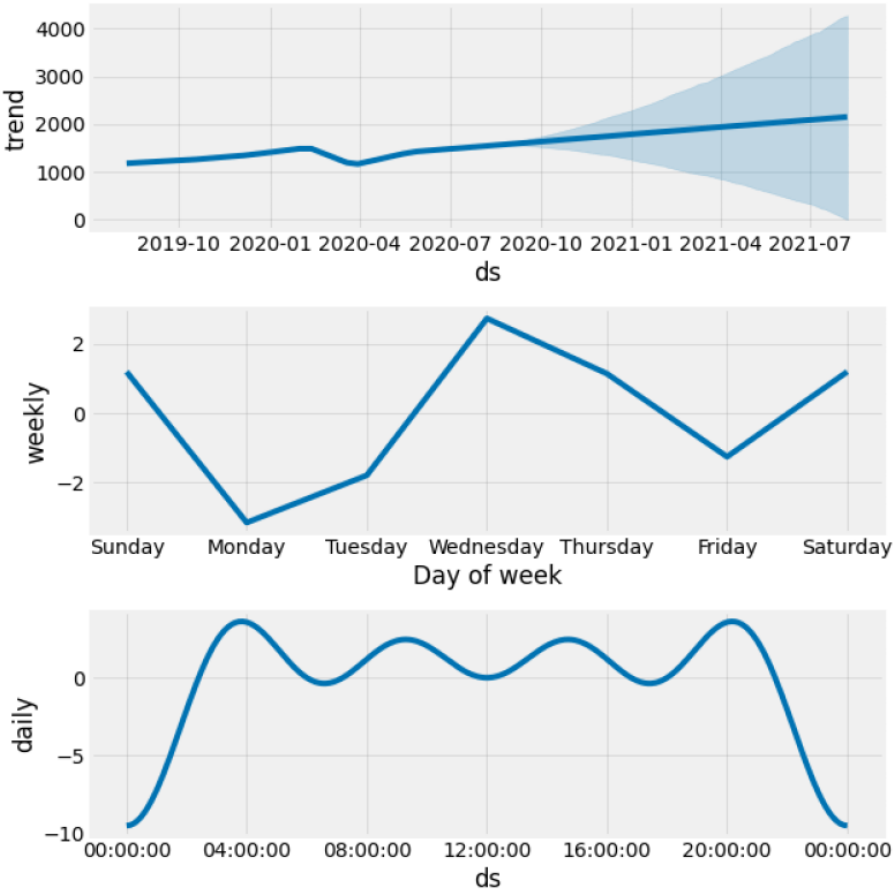
لقد نجحنافي ملاءمة البيانات مع نموذج Facebook Prophet. الآن دعنا نلقي نظرة على التنبؤ بسعر السهم الذي قدمه النموذج:

```
future = m.make_future_dataframe(periods=365)
predictions=m.predict(future)
m.plot(predictions)
plt.title("Prediction of GOOGLE Stock Price")
plt.xlabel("Date")
plt.ylabel("Closing Stock Price")
plt.show()
```



دعنا الآن نلقي نظرة على التأثيرات الموسمية على هذا التنبؤ الذي صنعه نموذجنا:

```
m.plot_components(predictions)
plt.show()
```



أتمنى أن تكون قد أحببت هذا المقال عن التنبؤ بأسعار الأسهم باستخدام نموذج Facebook Prophet.

المصدر:

https://thecleverprogrammer.com/2020/08/09/stock-price-prediction-/with-facebook-prophet-model

14) نموذج ARIMA في التعلم الآلي ARIMA Model in Machine Learning

نموذج ARIMA يعني متوسط متحرك متكامل ذاتي الانحدار Autoregressive Integrated Moving Average. يوفر هذا النموذج مجموعة من الدوال التي تعتبر قوية جدًا ومرنة لأداء أي مهمة تتعلق بالتنبؤ بالسلاسل الزمنية Time Series Forecasting.

في التعلم الآلي، يعد نموذج ARIMA عمومًا فئة من النماذج الإحصائية التي تعطي مخرجات تعتمد خطيًا على قيمها السابقةفي مجموعة من العوامل العشوائية.

أثناء اختيار نموذج التنبؤ بالسلاسل الزمنية المناسبة، نحتاج إلى تصوير البيانات لتحليل الاتجاهات والفصول الموسمية والدورات. عندما تكون الموسمية ميزة قوية جدًا للسلسلة الزمنية، نحتاج إلى التفكيرفي نموذج مثل ARIMA الموسمي (SARIMA).

يعمل نموذج ARIMA باستخدام نموذج التأخر الموزع distributed lag الذي تستخدم فيه الخوارزميات للتنبؤ بالمستقبل بناءً على القيم المتأخرة lagged values.في هذه المقالة، سأوضح لك كيفية استخدام نموذج ARIMA باستخدام مثال عملي جدًافي التعلم الآلي وهو اكتشاف الشذوذ Anomaly Detection.

كشف الشذوذ باستخدام نموذج ARIMA

الكشف عن الشذوذ Anomaly Detection يعني تحديد الأحداث غير المتوقعةفي العملية. يعني اكتشاف التهديدات التي تتعرض لها أنظمتنا والتي قد تسبب ضررًا من حيث الأمان وتسرب المعلومات المهمة.

لا تقتصر أهمية "كشف الشذوذ" على الأمان، بل يتم استخدامه للكشف عن أي حدث لا يتوافق مع توقعاتنا. سأشرح لك هنا كيف يمكننا استخدام نموذج ARIMA لاكتشاف الشذوذ.

سأستخدم البيانات التي تستند إلى مقاييس الدقيقة لاستخدام وحدة المعالجة المركزية للمضيف. لنبدأ الآن بهذه المهمة عن طريق استيراد المكتبات الضرورية:

```
import pandas as pd
!pip install pyflux
import pyflux as pf
from datetime import datetime
```

الآن دعنا نستورد البيانات ونلقي نظرة سريعة على البيانات وبعض الأفكار الخاصة بها. يمكنك تنزيل البيانات التي أستخدمهافي هذه المهمة من هنا.

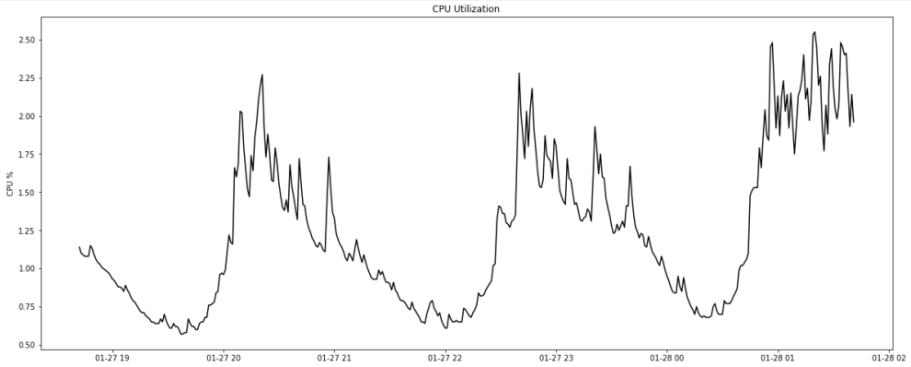
```
from google.colab import files
uploaded = files.upload()
```

```
data_train_a = pd.read_csv('cpu-train-a.csv', parse_dates=[0],
infer_datetime_format=True)
data_test_a = pd.read_csv('cpu-test-a.csv', parse_dates=[0],
infer_datetime_format=True)
data_train_a.head()
```

	datetime	cpu
0	2017-01-27 18:42:00	1.14
1	2017-01-27 18:43:00	1.10
2	2017-01-27 18:44:00	1.09
3	2017-01-27 18:45:00	1.08
4	2017-01-27 18:46:00	1.08

الآن، دعنا نرسم هذه البيانات لإلقاء نظرة سريعة على ما نعمل معه:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(20,8))
plt.plot(data_train_a['datetime'], data_train_a['cpu'],
color='black')
plt.ylabel('CPU %')
plt.title('CPU Utilization')
```



استخدام نموذج ARIMA

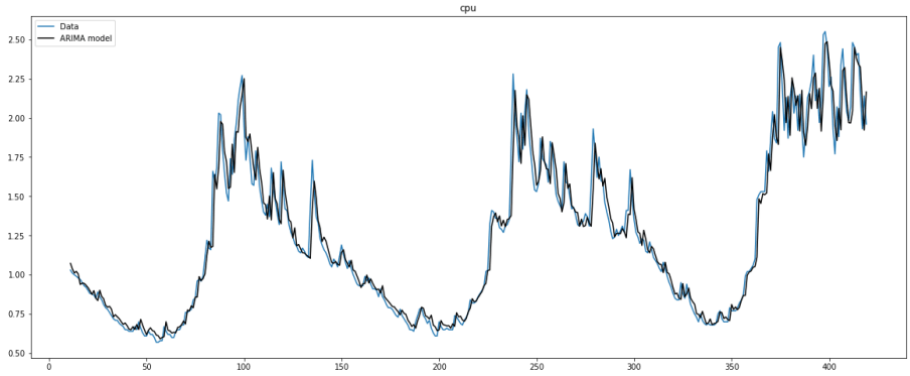
الآن، دعنا نرى كيف يمكننا استخدام نموذج ARIMA للتنبؤ بالبيانات:

```
model_a = pf.ARIMA(data=data_train_a, ar=11, ma=11, integ=0,
target='cpu')
x = model_a.fit("M-H")
```

```
Acceptance rate of Metropolis-Hastings is 0.0
Acceptance rate of Metropolis-Hastings is 0.026
Acceptance rate of Metropolis-Hastings is 0.2346
Tuning complete! Now sampling.
Acceptance rate of Metropolis-Hastings is 0.244425
```

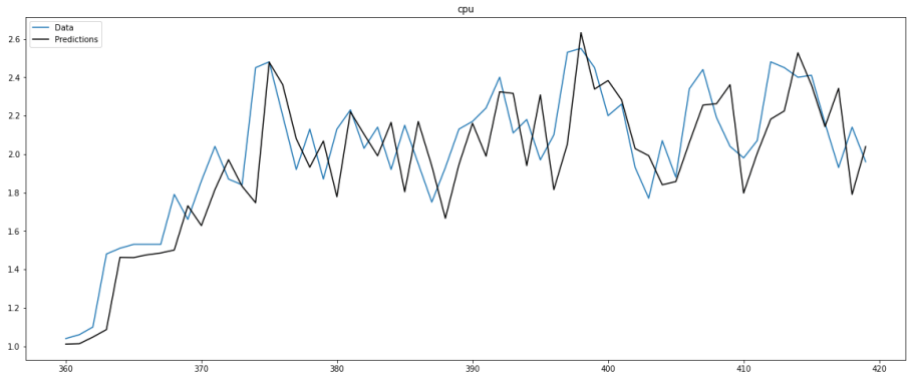
الآن، دعونا نرسم نموذجنا:

```
model_a.plot_fit(figsize=(20,8))
```



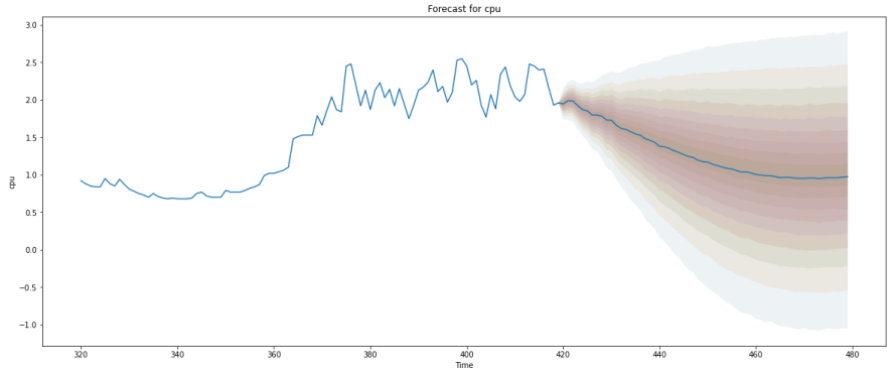
يوضح الإخراج أعلاه استخدام وحدة المعالجة المركزية بمرور الوقت مع تنبؤ نموذج ARIMA. الآن دعنا نجري اختبارًا نموذجيًا لتقييم أداء نموذجنا:

```
model_a.plot_predict_is(h=60, figsize=(20,8))
```



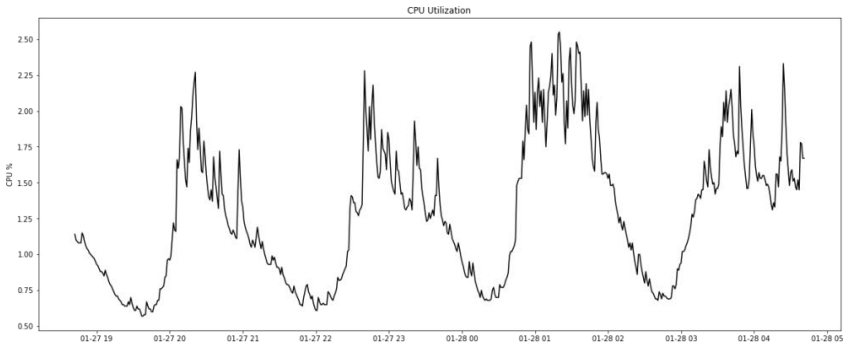
يُظهر الإخراج أعلاهفي العينة (مجموعة التدريب) لنموذج التنبؤ ARIMA الخاص بنا. الآن، سأقوم بتشغيل التنبؤ الفعلي actual prediction، باستخدام أحدث 100 نقطة بيانات تمت ملاحظتها متبوعة بـ 60 نقطة متوقعة:

```
model_a.plot_predict(h=60,past_values=100,figsize=(20,8))
```



دعونا نجري نفس اكتشاف الشذوذ على شريحة أخرى من مجموعة بيانات استخدام وحدة المعالجة المركزية التي تم التقاطهافي وقت مختلف:

```
data_train_b = pd.read_csv('cpu-train-b.csv', parse_dates=[0],
infer_datetime_format=True)
data_test_b = pd.read_csv('cpu-test-b.csv', parse_dates=[0],
infer_datetime_format=True)
plt.figure(figsize=(20,8))
plt.plot(data_train_b['datetime'], data_train_b['cpu'],
color='black')
plt.ylabel('CPU %')
plt.title('CPU Utilization')
```



الآن، دعونا نلائم (fit) هذه البيانات على النموذج:

```
model_b = pf.ARIMA(data=data_train_b, ar=11, ma=11, integ=0,
target='cpu')
x = model_b.fit("M-H")
```

Acceptance rate of Metropolis-Hastings is 0.0

Acceptance rate of Metropolis-Hastings is 0.016

Acceptance rate of Metropolis-Hastings is 0.1344

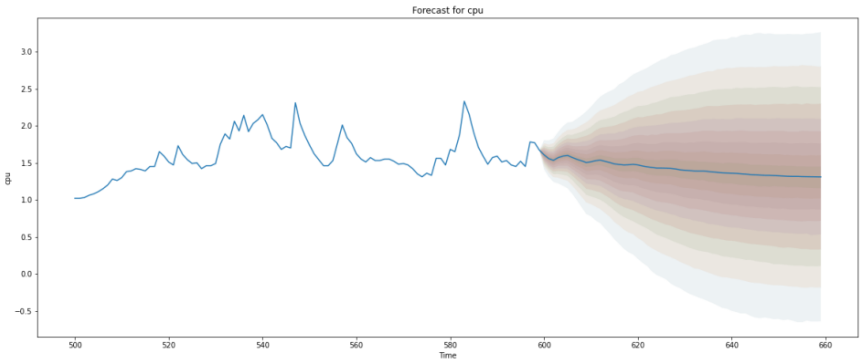
Acceptance rate of Metropolis-Hastings is 0.21025

Acceptance rate of Metropolis-Hastings is 0.23585

Tuning complete! Now sampling.

Acceptance rate of Metropolis-Hastings is 0.34395

```
model_b.plot_predict(h=60,past_values=100,figsize=(20,8))
```



يمكننا رسم الحالة الشاذة التي تحدث بعد فترة قصيرة من فترة التدريب، حيث تقع القيم المرصودة ضمن نطاقات منخفضة الثقة، لذلك سترفع تنبيهًا عن الحالة الشاذة.

أتمنى أن تكون قد أحببت هذا المقال عن اكتشاف الشذوذ باستخدام نموذج ARIMA.

المصدر:

https://thecleverprogrammer.com/2020/08/04/arima-model-in-machine-learning/

15) تحليل السلاسل الزمنية والتنبؤ بها باستخدام بايثون Time Series Analysis and Forecasting with Python

يحمل تحليل السلاسل الزمنية Time Series Analysis طرقًا للبحث في إحصائيات السلاسل الزمنية time-series statistics لاستخراج السمات الإحصائية من البيانات. يستخدم التنبؤ بالسلاسل الزمنية Time Series Forecasting في تدريب نموذج التعلم الآلي للتنبؤ بالقيم المستقبلية باستخدام الأهمية التاريخية.

يستخدم تحليل السلاسل الزمنية على نطاق واسع في تدريب نماذج التعلم الآلي للاقتصاد، والتنبؤ بالطقس، والتنبؤ بأسعار الأسهم، بالإضافة إلى التنبؤ بالمبيعات.

يمكن القول إن تحليل السلاسل الزمنية يستخدم على نطاق واسع في الحقائق القائمة على الميزات غير الثابتة non-stationary features.

تحليل السلاسل الزمنية والتنبؤ باستخدام بايثون

في هذه المقالة، سأستخدم طرقًا مختلفة للتنبؤ بالمبيعات باستخدام تحليل السلاسل الزمنية باستخدام بايثون. يمكنك تنزيل مجموعة البيانات التي استخدمتها في هذه المقالة أدناه.

- تحميل مجموعة البيانات.

لنبدأ بهذا البرنامج التعليمي حول التنبؤ بالسلسلة الزمنية باستخدام بايثون عن طريق استيراد المكتبات.

```
import warnings
import itertools
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
import pandas as pd
import statsmodels.api as sm
import matplotlib
matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['xtick.labelsize'] = 12
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'
```

هناك فئات مختلفة في مجموعة البيانات، تتيح لك البدء من تحليل السلاسل الزمنية والتنبؤ بمبيعات الأثاث.

```
df = pd.read_excel("Superstore.xls")
furniture = df.loc[df['Category'] == 'Furniture']
furniture['Order Date'].min(), furniture['Order Date'].max()
```

```
Timestamp('2014-01-06 00:00:00'), Timestamp('2017-12-30
00:00:00')
```

معالجة البيانات

تتضمن المعالجة المسبقة للبيانات إزالة الأعمدة التي لا نحتاج إليها، والبحث عن القيم المفقودة، وما إلى ذلك.

```
cols = ['Row ID', 'Order ID', 'Ship Date', 'Ship Mode',
        'Customer ID', 'Customer Name',
        'Segment', 'Country', 'City', 'State',
        'Postal Code', 'Region', 'Product ID',
        'Category', 'Sub-Category', 'Product Name',
        'Quantity', 'Discount', 'Profit']
furniture.drop(cols, axis=1, inplace=True)
furniture = furniture.sort_values('Order Date')
furniture.isnull().sum()
```

```
Order Date    0
Sales         0
dtype: int64
```

```
furniture = furniture.groupby('Order
Date')['Sales'].sum().reset_index()
```

فهرسة بيانات السلاسل الزمنية

```
furniture = furniture.set_index('Order Date')
furniture.index
```

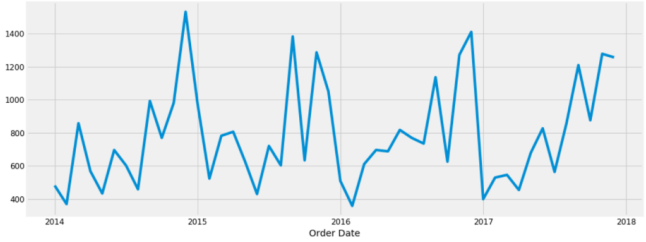
```
DatetimeIndex(['2014-01-06', '2014-01-07', '2014-01-10', '2014-01-11',
               '2014-01-13', '2014-01-14', '2014-01-16', '2014-01-19',
               '2014-01-20', '2014-01-21',
               ...
               '2017-12-18', '2017-12-19', '2017-12-21', '2017-12-22',
               '2017-12-23', '2017-12-24', '2017-12-25', '2017-12-28',
               '2017-12-29', '2017-12-30'],
              dtype='datetime64[ns]', name='Order Date', length=889, freq=None)
```

يبدو العملفي DateTime الحالي صعبًا بعض الشيءفي العمل ضمن مجموعة البيانات، لذلك سأستخدم سعر المبيعات اليوميةفي متوسط الشهر للحفاظ على بساطتها. سأستخدم بداية كل شهر كطابع زمني.

```
y = furniture['Sales'].resample('MS').mean()
```

تصوير بيانات مبيعات الأثاث

```
y.plot(figsize=(15, 6))
plt.show()
```

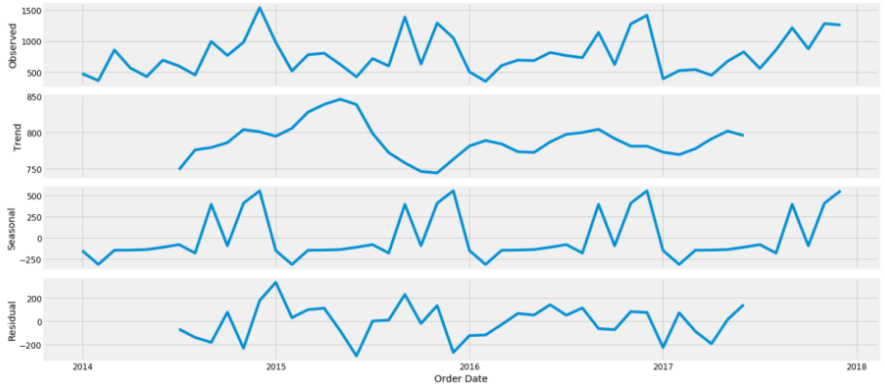


يمكن استخلاص بعض الأنماط من الشكل أعلاه، فالسلسلة الزمنية منقوشة بشكل موسمي seasonally مثل المبيعات منخفضةفي بداية كل عام، وتزداد المبيعاتفي نهاية العام.

الآن دعنا نرسم هذه البيانات باستخدام طريقة تحليل السلاسل الزمنية التي ستسمح للسلسلة الزمنية لدينا بالتحلل إلى ثلاث مكونات:

1. اتجاه Trend.
2. موسم Season.
3. ضوضاء Noise.

```
from pylab import rcParams
rcParams['figure.figsize'] = 18, 8
decomposition = sm.tsa.seasonal_decompose(y, model='additive')
fig = decomposition.plot()
plt.show()
```



يوضح الشكل أعلاه أن مبيعات الأثاث غير مستقرة بسبب المواسم.

توقع السلاسل الزمنية مع ARIMA

ARIMA هي واحدة من أكثر الطرق استخدامًافي التنبؤ بالسلاسل الزمنية. ARIMA اختصار لـ Autoregressive Integrated Moving Average. الآن سأستخدم طريقة ARIMAفي عملية أخرى للتنبؤ بالسلسلة الزمنية.

```
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in
list(itertools.product(p, d, q))]
print('Examples of parameter combinations for Seasonal
ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
```

```
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))
```

```
Examples of parameter combinations for Seasonal ARIMA...
SARIMAX: (0, 0, 1) x (0, 0, 1, 12)
SARIMAX: (0, 0, 1) x (0, 1, 0, 12)
SARIMAX: (0, 1, 0) x (0, 1, 1, 12)
SARIMAX: (0, 1, 0) x (1, 0, 0, 12)
```

هذه الخطوة هي عملية اختيار المعلماتفي نموذج تنبؤ السلاسل الزمنية لمبيعات الأثاث.

```
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(y,
                                            order=param,

seasonal_order=param_seasonal,

enforce_stationarity=False,

enforce_invertibility=False)
results = mod.fit()
print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal,
results.aic))
        except:
            continue
```

```
ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - AIC:1131.2657078645939
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:497.23144334183365
ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:1001.3915524374769
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:318.0047199116341
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:720.9252270758095
ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:2876.7174897071977
ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:466.56074298091255
ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:499.54290594685824
ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:2461.517421827548
ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:319.98848769468657
ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:1287.5697512865586
ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:497.78896630044073
ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:1388.8924232046936
ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:319.7714068109211
ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:649.9056176816999
ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:3307.7208814993064
ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:458.8705548482932
ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:486.18329774427826
ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:2625.602326434297
ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:310.75743684172994
ARIMA(1, 0, 0)x(0, 0, 0, 12)12 - AIC:692.1645522067712
ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:1399.3709974017943
ARIMA(1, 0, 0)x(0, 1, 0, 12)12 - AIC:479.46321478521355
ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:480.92593679352177
ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:1431.0752736869172
ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:304.4664675084554
ARIMA(1, 0, 1)x(0, 0, 0, 12)12 - AIC:665.779444218685
ARIMA(1, 0, 1)x(0, 0, 1, 12)12 - AIC:246116.34689777798
ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:468.3685195814987
ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:482.5763323876739
ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:3365796.8535189056
ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:306.0156002122138
ARIMA(1, 1, 0)x(0, 0, 0, 12)12 - AIC:671.2513547541902
ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:1393.2157168383435
ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:479.2003422281134
ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:475.34036587848493
ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:2102.468501404909
ARIMA(1, 1, 0)x(1, 1, 0, 12)12 - AIC:300.6270901345443
ARIMA(1, 1, 1)x(0, 0, 0, 12)12 - AIC:649.0318019835024
ARIMA(1, 1, 1)x(0, 0, 1, 12)12 - AIC:2603.9208285600357
ARIMA(1, 1, 1)x(0, 1, 0, 12)12 - AIC:460.4762687610111
ARIMA(1, 1, 1)x(1, 0, 0, 12)12 - AIC:469.52503546608614
ARIMA(1, 1, 1)x(1, 0, 1, 12)12 - AIC:2586.7750340396897
ARIMA(1, 1, 1)x(1, 1, 0, 12)12 - AIC:297.7875439553055
```

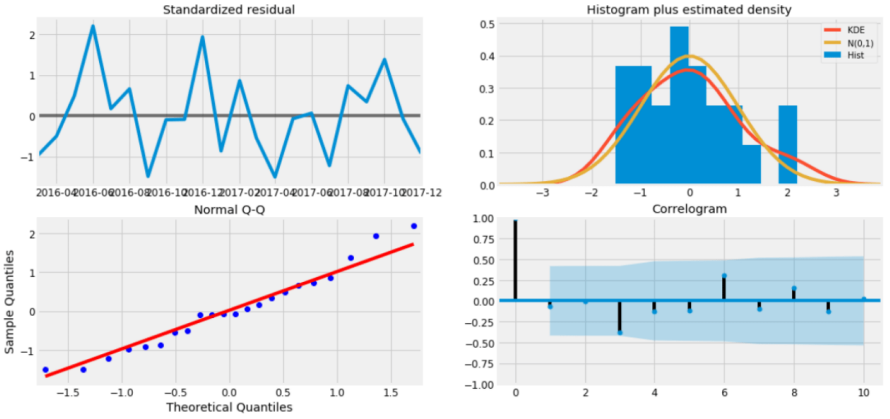
تطبيق نموذج ARIMA

```
mod = sm.tsa.statespace.SARIMAX(y,
                                order=(1, 1, 1),
                                seasonal_order=(1, 1, 0, 12),
                                enforce_stationarity=False,
                                enforce_invertibility=False)
results = mod.fit()
print(results.summary().tables[1])
```

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.0146      0.342      0.043      0.966      -0.655       0.684
ma.L1         -1.0000      0.360     -2.781      0.005      -1.705      -0.295
ar.S.L12      -0.0253      0.042     -0.609      0.543      -0.107       0.056
sigma2      2.958e+04   1.22e-05   2.43e+09      0.000    2.96e+04    2.96e+04
==============================================================================
```

الآن سأقوم بتشغيل نموذج التشخيص Model diagnosis؛ يعد إجراء تشخيص نموذجي أمرًا ضروريًافي تنبؤ السلاسل الزمنية للتحقيقفي أي سلوك غير عاديفي النموذج.

```
results.plot_diagnostics(figsize=(16, 8))
plt.show()
```



التحقق من صحة توقعات السلاسل الزمنية

لفهم دقة نموذج التنبؤ بالسلسلة الزمنية لدينا، سأقارن المبيعات المتوقعة predicted sales بالمبيعات الفعلية actual sales، وسأحدد التوقعات لتبدأفي 2017-01-01 حتى نهاية مجموعة البيانات.

```
pred = results.get_prediction(start=pd.to_datetime('2017-01-
01'), dynamic=False)
pred_ci = pred.conf_int()
ax = y['2014':].plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='One-step ahead
Forecast', alpha=.7, figsize=(14, 7))
ax.fill_between(pred_ci.index,
```

```
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha=.2)
ax.set_xlabel('Date')
ax.set_ylabel('Furniture Sales')
plt.legend()
plt.show()
```

يوضح الشكل أعلاه القيم المرصودة observed values مقارنة بالتنبؤات المتوقعة forecast predictions. تتوافق الصورة مع المبيعات الفعلية، بشكل جيد حقًا، والتي تُظهر تحولًا تصاعديًافي البداية وتلتقط الموسميةفي نهاية العام.

```
y_forecasted = pred.predicted_mean
y_truth = y['2017-01-01':]
mse = ((y_forecasted - y_truth) ** 2).mean()
print('The Mean Squared Error of our forecasts is
{}'.format(round(mse, 2)))
```

متوسط الخطأ التربيعي Mean Squared Error لتوقعاتنا هو 22993.58.

```
print('The Root Mean Squared Error of our forecasts is
{}'.format(round(np.sqrt(mse), 2)))
```

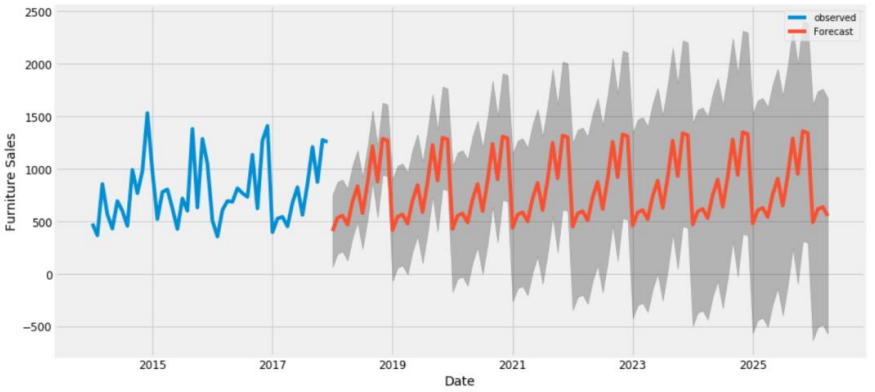
الخطأ التربيعي لمتوسط الجذر لتوقعاتنا هو 151.64

في الإحصاء، يقيس متوسط الخطأ التربيعي Mean Squared Error (MSE) للمقدر متوسط average مربعات الخطأ، أي التمييز التربيعي المشترك بين القيم المتوقعة وما هو مقدر. يعد MSE مقياسًا لغرامة المقدر، وأميالها غير سالبة باستمرار، وكلما كان MSE أصغر، كلما اقتربنا من تحديد الطريق المناسب بشكل ممتاز.

يخبرنا Root Mean Square Error (RMSE) أن نسختنا كانت قادرة على التنبؤ بمتوسط دخل الأثاث اليوميفي مجموعة الاختبار ضمن 151.64 من الدخل الفعلي. يتراوح دخل أثاثنا اليومي من حوالي 400 إلى أكثر من 1200.في رأيي، هذا إصدار جيد جدًا حتى الآن.

إنتاج وتصوير التنبؤات

```
pred_uc = results.get_forecast(steps=100)
pred_ci = pred_uc.conf_int()
ax = y.plot(label='observed', figsize=(14, 7))
pred_uc.predicted_mean.plot(ax=ax, label='Forecast')
ax.fill_between(pred_ci.index,
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha=.25)
ax.set_xlabel('Date')
ax.set_ylabel('Furniture Sales')
plt.legend()
plt.show()
```



من دون شك، استحوذ نموذج تنبؤات السلسلة الزمنية لدينا على موسمية أرباح الأثاث. بينما نتوقع المزيدفي المستقبل، من الطبيعي جداً أن نصبح أقل ثقةفي قيمنا. ينعكس هذا من خلال فترات الإيمان بالذات الناتجة عن نموذجنا، والتي تزداد أهمية كلما انتقلنا بالمثل إلى المستقبل.

المصدر:

https://thecleverprogrammer.com/2020/07/01/time-series-analysis-and-/forecasting-with-python

16) مشروع علم البيانات في السلاسل الزمنية Data Science Project on Time Series

كمثال للعمل مع بعض بيانات السلاسل الزمنية time series data، دعنا نلقي نظرة على عدد الدراجاتفي فريمونت بريدجفي سياتل. تأتي هذه البيانات من عداد الدراجات الآلي، الذي تم تركيبهفي أواخر عام 2012، والذي يحتوي على أجهزة استشعار حثي على الأرصفة الشرقية والغربية للجسر. يمكن تنزيل عدد الدراجات بالساعة من هنا.

بمجرد تنزيل مجموعة البيانات هذه، يمكننا استخدام Pandas لقراءة إخراج CSVفي DataFrame. سنحدد أننا نريد التاريخ Date كفهرس index، ونريد أن يتم تحليل هذه التواريخ تلقائيًا:

```
import pandas as pd
data = pd.read_csv("fremont-bridge.csv", index_col= 'Date',
parse_dates=<strong>True</strong>)
data.head()
```

	Fremont Bridge West Sidewalk	Fremont Bridge East Sidewalk
Date		
2012-10-03 00:00:00	4.0	9.0
2012-10-03 01:00:00	4.0	6.0
2012-10-03 02:00:00	1.0	1.0
2012-10-03 03:00:00	2.0	3.0
2012-10-03 04:00:00	6.0	1.0

لتسهيل الأمر، سنعالج مجموعة البيانات هذه بشكل أكبر عن طريق تقصير أسماء الأعمدة وإضافة عمود الاجمالي "Total":

```
data.columns = ["West", "East"]
data["Total"] = data["West"] + data["East"]
data.head()
```

	West	East	Total
Date			
2012-10-03 00:00:00	4.0	9.0	13.0
2012-10-03 01:00:00	4.0	6.0	10.0
2012-10-03 02:00:00	1.0	1.0	2.0
2012-10-03 03:00:00	2.0	3.0	5.0
2012-10-03 04:00:00	6.0	1.0	7.0

دعنا الآن نلقي نظرة على ملخص الإحصائيات لهذه البيانات:

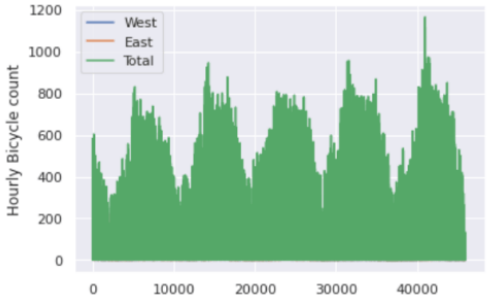
```
data.dropna().describe()
```

	West	East	Total
count	45976.000000	45976.000000	45976.000000
mean	54.712306	55.479315	110.191622
std	72.935797	80.232055	139.189603
min	0.000000	0.000000	0.000000
25%	7.000000	7.000000	15.000000
50%	29.000000	28.000000	60.000000
75%	72.000000	68.000000	144.000000
max	854.000000	717.000000	1165.000000

تصوير البيانات

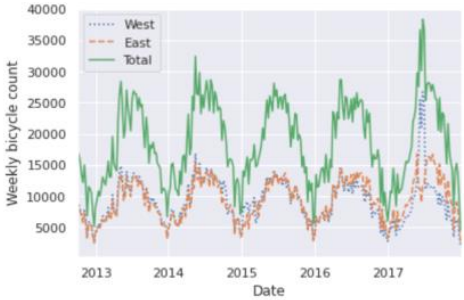
يمكننا الحصول على نظرة ثاقبة لمجموعة البيانات من خلال تصويرها. لنبدأ بتخطيط البيانات الأولية:

```
import matplotlib.pyplot as plt
import seaborn
seaborn.set()
data.plot()
plt.ylabel("Hourly Bicycle count")
plt.show()
```



25000 عينة كل ساعة تقريبًا كثيفة جدًا بالنسبة لنا لفهمها كثيرًا. يمكننا الحصول على مزيد من البصيرة عن طريق إعادة اخذ العينات resampling إلى شبكة أكثر خشونة coarser grid. دعنا نعيد أخذ العينة حسب الأسبوع:

```
weekly = data.resample("W").sum()
weekly.plot(style=[':', '--', '-'])
plt.ylabel('Weekly bicycle count')
plt.show()
```



يوضح لنا هذا بعض الاتجاهات الموسمية seasonal trends المثيرة للاهتمام: كما قد تتوقع، يركب الناس دراجات أكثر في الصيف أكثر من الشتاء، وحتى في موسم معين، يختلف استخدام الدراجات من أسبوع لآخر.

هناك طريقة أخرى مفيدة لتجميع البيانات وهي استخدام متوسط متحرك rolling mean، باستخدام دالة pd.rolling_mean(). سنقوم هنا بعمل متوسط متحرك لمدة 30 يومًا لبياناتنا، مع التأكد من توسيط النافذة:

```
daily = data.resample('D').sum()
daily.rolling(30, center=True).sum().plot(style=[':', '--', '-
'])
plt.ylabel('mean hourly count')
plt.show()
```

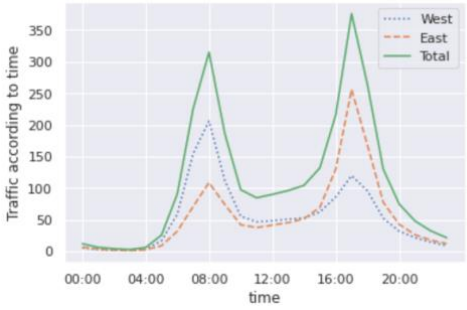
يرجع سبب خشونة النتيجة إلى القطع الصعب hard cutoff للنافذة. يمكننا الحصول على نسخة أكثر سلاسة من المتوسط المتحرك باستخدام دالة النافذة window function ـ على سبيل المثال، نافذة Gaussian.

```
daily.rolling(50, center=True,

win_type='gaussian').sum(std=10).plot(style(['-' ,'--',':']=
plt.show()
```

التنقيب في البيانات

في حين أن طرق عرض البيانات المتجانسة مفيدة للحصول على فكرة عن الاتجاه العامفي البيانات، فإنها تخفي الكثير من البُنية المثيرة للاهتمام. على سبيل المثال، قد نرغبفي النظر إلى متوسط عدد الزيارات كدالة للوقت من اليوم. يمكننا القيام بذلك باستخدام دالة GroupBy:

```
import numpy as np
by_time = data.groupby(data.index.time).mean()
hourly_ticks = 4 * 60 * 60 * np.arange(6)
by_time.plot(xticks= hourly_ticks, style=[':', '--', '-'])
plt.ylabel("Traffic according to time")
plt.show()
```



المصدر:

https://thecleverprogrammer.com/2020/05/08/data-science-project-on-time-series/

17) AutoTS في بايثون AutoTS in Python

AutoTS هي مكتبة تلقائية للتعلم الآليفي بايثون تم تطويرها لمهمة التنبؤ التلقائي بالسلاسل الزمنية automatic time series forecasting. يمكنك استخدام هذه المكتبة لأي مهمة تتعلق بالتنبؤ بالسلسلة الزمنية مثل التنبؤ بأسعار الأسهم لعدد n من الأيام التالية.في هذه المقالة، سوف آخذك خلال برنامج تعليمي حول مكتبة AutoTSفي بايثون.

ما هو AutoTS في بايثون؟

تعني AutoTS سلسلة زمنية تلقائية Automatic Time Series، وهي عبارة عن مكتبة للتعلم الآليفي بايثون تُستخدم لمهمة التنبؤ بالسلسلة الزمنية. لقد استخدمته مؤخرًا للتنبؤ بأسعار أسهم Apple للأيام العشرة القادمة وكانت الأرقام الناتجة دقيقة للغاية. بعض الميزات الشائعة لمكتبة AutoTSفي بايثون هي:

1. يمكن استخدامه للعثور على أفضل نموذج للتنبؤ بالسلاسل الزمنية والذي يعتمد على نوع البيانات التي تستخدمها.
2. يمكنه التعامل مع كل من السلاسل الزمنية أحادية المتغير univariate ومتعددة المتغيرات multivariate.
3. يمكنه أيضًا التعامل مع البيانات الفوضوية messy data عن طريق إزالة قيم NaN وتعبئتها ويمكنه أيضًا التعامل مع القيم المتطرفة outliers.
4. يمكنك استخدام النماذج التي توفرها مكتبة بايثون هذه للنشر أيضًا.

كانت هذه بعض الميزات المهمة لمكتبة AutoTSفي بايثون. يحتوي على المزيد من الميزات للتنبؤ بالسلسلة الزمنية، يمكنك استكشاف المزيد حول هذه المكتبة من وثائقها الرسمية من هنا. في القسم أدناه، سأأخذك خلال برنامج تعليمي حول مكتبة AutoTSفي بايثون لمهمة التنبؤ بسعر السهم.

AutoTS في بايثون (تعليمي)

إذا لم تستخدم مكتبة بايثون هذه مطلقًافي مهمة التنبؤ بالسلسلة الزمنية من قبل ، فيمكنك تثبيتها بسهولةفي نظامك باستخدام الأمر pip ؛ `pip install autots` . دعنا الآن نرى كيفية استخدام مكتبة AutoTSفي بايثون لمهمة توقع أسعار الأسهم. سأبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("AAPL.csv")
```

في هذا البرنامج التعليمي، أستخدم بيانات أسعار أسهم Apple التي تم تنزيلها من Yahoo Finance. الآن قبل استخدام مكتبة AutoTs لمهمة التنبؤ بسعر السهم، دعنا نجهز البيانات ونلقي نظرة على أسعار إغلاق أسهم Apple:

```
data = data[["Date", "Close"]]
data["Date"] = pd.to_datetime(data.Date)
data["Close"].plot(figsize=(12, 8), title="Apple Stock
Prices", fontsize=20, label="Close Price")
plt.legend()
plt.grid()
plt.show()
```

الآن دعنا نرى كيفية استخدام مكتبة AutoTS لمهمة التنبؤ التلقائي للسلاسل الزمنية:

```
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)
```

في قسم الكود أعلاه، قمت بتعيين معامل طول التوقعات forecast_length على 10، وهو ما يعني الفترة التي نريد توقع أسعار الأسهم فيها. لذلكفي الإخراج، سوف نحصل على أسعار إغلاق سهم Apple للأيام العشرة القادمة. لذلك دعونا نستخدم دالة التنبؤ predict function ونلقي نظرة على الناتج:

```
prediction = model.predict()
forecast = prediction.forecast
print("Stock Price Prediction of Apple")
print(forecast)
```

```
Stock Price Prediction of Apple
                Close
2021-04-19  132.794042
2021-04-20  132.882870
2021-04-21  132.971698
2021-04-22  133.060526
2021-04-23  133.149354
2021-04-26  133.415838
2021-04-27  133.504666
2021-04-28  133.593494
2021-04-29  133.682322
2021-04-30  133.771150
```

الملخص

تعني AutoTS سلاسل زمنية تلقائية، وهي مكتبة AutoMLفي بايثون يمكن استخدامها لكل مهمة من مهام التنبؤ بالسلسلة الزمنية. أتمنى أن تكون هذه المقالة قد أعجبتكفي برنامج تعليمي في مكتبة AutoTSفي بايثون.

المصدر:

https://thecleverprogrammer.com/2021/04/19/autots-in-python-/tutorial

18) مخطط السلاسل الزمنية باستخدام بايثون Time Series Graph using Python

الرسم البياني للسلسلة الزمنية هو مخطط خطي line plot يعرض الاتجاهات أو الأنماط عبر مجموعة بيانات مجمعة خلال فترة زمنية. على سبيل المثال، عندما تتخيل مخططًا خطيًا للمبيعات اليومية التي تقوم بها شركة ما، فإنك تصور رسمًا بيانيًا لسلسلة زمنية. إنها واحدة من أهم تصويرات البيانات لكل عالم بيانات. لذلك إذا كنت تريد معرفة كيفية تصويرها باستخدام بايثون، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك من خلال برنامج تعليمي حول تصوير رسم بياني للسلسلة الزمنية باستخدام بايثون.

مخطط السلاسل الزمنية

مخطط السلسلة الزمنية هو مخطط خطي يستخدم لتصوير بيانات السلاسل الزمنية. بيانات السلاسل الزمنية هي البيانات التي يتم جمعها خلال فترة زمنية. بعض الأمثلة الشائعة لبيانات السلاسل الزمنية هي بيانات أسعار الأسهم، وبيانات المبيعات اليومية، وبيانات حالات فيروس كورونا اليومية، وما إلى ذلك.

أثناء تصوير مخطط السلسلة الزمنية، تقع الفواصل الزمنية time intervals على المحور السيني، وتقع نقاط البيانات data points التي تم جمعها خلال تلك الفترات الزمنية على المحور الصادي. على سبيل المثال، أثناء تصوير مخطط السلسلة الزمنية لبيانات أسعار الأسهم، سيقع عمود التاريخ date column على المحور السيني، وسيقع عمود السعر price column على المحور الصادي.

أتمنى أن تكون قد فهمت الآن ماهية مخطط السلسلة الزمنية ولماذا يتم استخدامهفي علم البيانات. الآن،في القسم أدناه، سوف أطلعك على كيفية تصوير مخطط السلسلة الزمنية باستخدام بايثون.

مخطط السلسلة الزمنية باستخدام بايثون

لتصوير مخطط السلسلة الزمنية باستخدام بايثون، سأستخدم مجموعة بيانات أسعار الأسهم. هناك العديد من المكتباتفي بايثون لتصوير البيانات؛ سأستخدم Plotly لأنه من السهل تصوير التصورات التفاعلية باستخدام plotly. فلنبدأ هذه المهمة بجمع 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=360)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2

data = yf.download('AAPL,'
          start=start_date,
          end=end_date,
          progress=False(
print(data.head())
```

```
                 Open        High  ...   Adj Close     Volume
Date                               ...
2021-01-19  127.779999  128.710007  ...  127.046783   90757300
2021-01-20  128.660004  132.490005  ...  131.221054  104319500
2021-01-21  133.800003  139.669998  ...  136.031372  120150900
2021-01-22  136.279999  139.850006  ...  138.217926  114459400
2021-01-25  143.070007  145.089996  ...  142.044327  157611700

[5 rows x 6 columns]
```

يتم جمع البيانات الواردة أعلاه باستخدام yfinance API. يمكنك معرفة المزيد عنها من هنا.
الآن فيما يلي كيف يمكنك تصوير مخطط السلسة البيانية باستخدام بايثون:

```
import plotly.express as px
figure = px.line(data, x = data.index, y = "Close")
figure.show()
```

الملخص

مخطط السلسلة الزمنية هو مخطط خطي يستخدم لتصوير بيانات السلاسل الزمنية. بيانات السلاسل الزمنية هي البيانات التي يتم جمعها خلال فترة زمنية. أثناء تصور مخطط السلسلة الزمنية، تقع الفواصل الزمنية على المحور السيني، وتقع نقاط البيانات التي تم جمعها خلال تلك الفترات الزمنية على المحور الصادي. آمل أن تكون قد أحببت هذه المقالة حول تصور مخطط السلسلة الزمنية باستخدام بايثون

المصدر:

https://thecleverprogrammer.com/2022/01/12/time-series-graph-using-/python

19) تحليل سوق الأسهم باستخدام بايثون Stock Market Analysis using Python

يعني تحليل سوق الأسهم (Stock Market Analysis) تحليل الاتجاهات الحالية والتاريخية في سوق الأوراق المالية لاتخاذ قرارات البيع والشراء المستقبلية. يعد تحليل سوق الأوراق المالية أحد أفضل حالات استخدام علم البياناتفي التمويل. لذا، إذا كنت تريد تعلم كيفية تحليل سوق الأوراق المالية، فهذه المقالة مناسبة لك.في هذه المقالة، سأطلعك على مهمة تحليل سوق الأسهم باستخدام بايثون.

تحليل سوق الأسهم باستخدام بايثون

لتحليل سوق الأسهم، سأجمع بيانات أسعار أسهم Google.في نهاية هذه المقالة، ستتعلم تحليل سوق الأوراق المالية بشكل تفاعلي باستخدام لغة برمجة بايثون. لنبدأ بجمع بيانات أسعار أسهم Google. سأستخدم yfinance API من Yahoo Finance لجمع بيانات أسعار الأسهم. يمكنك معرفة المزيد عن API هنا.

إليك الآن كيفية جمع بيانات أسعار أسهم Google:

```
import pandas as pd
import yfinance as yf
import datetime
from datetime import date, timedelta
import plotly.graph_objects as go
import plotly.express as px

today = date.today()

d1 = today.strftime("%Y-%m-%d")
end_date = d1
d2 = date.today() - timedelta(days=365)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2

data = yf.download('GOOG',
                   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())
```

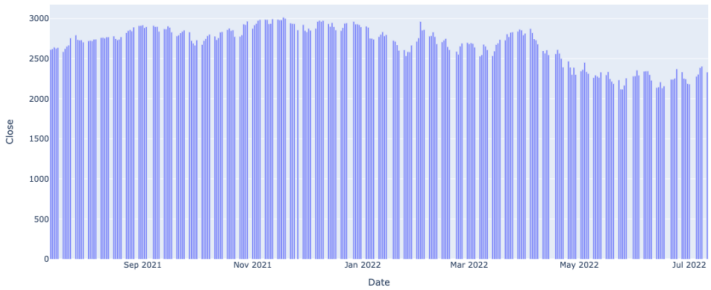
```
        Date         Open         High          Low        Close    Adj Close  \
0 2021-07-12  2596.669922  2615.399902  2592.000000  2611.280029  2611.280029
1 2021-07-13  2617.629883  2640.840088  2612.739990  2619.889893  2619.889893
2 2021-07-14  2638.030029  2659.919922  2637.959961  2641.649902  2641.649902
3 2021-07-15  2650.000000  2651.899902  2611.959961  2625.330078  2625.330078
4 2021-07-16  2632.820068  2643.659912  2616.429932  2636.909912  2636.909912

   Volume
0  847200
1  830900
2  895600
3  829300
4  742800
```

كلما قمت بتحليل سوق الأسهم، ابدأ دائمًا بمخطط الشموع (candlestick chart). مخطط الشموع هو أداة مفيدة لتحليل تحركات الاسعار لأسعار الأسهم. إليك كيفية تصوير مخطط الشموع لأسعار أسهم Google:

```
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 = "Google Stock Price Analysis",
xaxis_rangeslider_visible=False)
figure.show()
```

المخطط الشريطي (bar plot) هو أيضًا تصوير مفيد لتحليل سوق الأوراق المالية، على وجه التحديد على المدى الطويل. إليك كيفية تصوير أسعار إغلاق أسهم Google باستخدام مخطط شريطي:

```
figure = px.bar(data, x = "Date", y= "Close")
figure.show()
```



يعد شريط تمرير النطاق (range slider) أحد الأدوات القيمة لتحليل سوق الأوراق المالية. يساعدك على تحليل سوق الأوراق المالية بين نقطتين محددتين من خلال تحديد الفترة الزمنية بشكل تفاعلي. إليك كيفية إضافة شريط تمرير النطاق لتحليل سوق الأسهم:

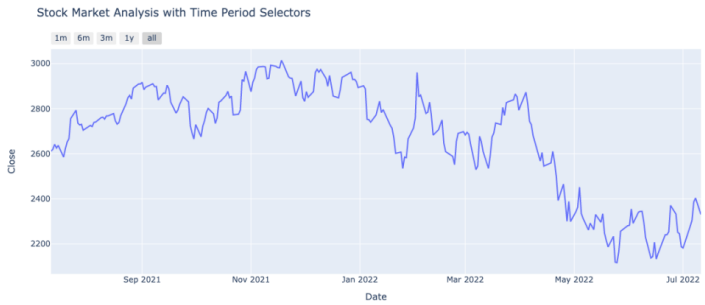
```
figure = px.line(data, x='Date', y='Close',
                 title='Stock Market Analysis with
Rangeslider')
figure.update_xaxes(rangeslider_visible=True)
figure.show()
```

استخدم شريط تمرير النطاق لتحليل سوق الأوراق المالية بشكل تفاعلي بين نقطتين

ميزة تفاعلية أخرى يمكنك إضافتها لتحليل سوق الأسهم هي محددات الفترة الزمنية (time period selectors). محددات الفترة الزمنية هي مثل الأزرار التي تعرض لك الرسم البياني لفترة زمنية محددة. على سبيل المثال، سنة، ثلاثة أشهر، ستة أشهر، إلخ. إليك كيفية إضافة أزرار لاختيار الفترة الزمنية لتحليل سوق الأسهم:

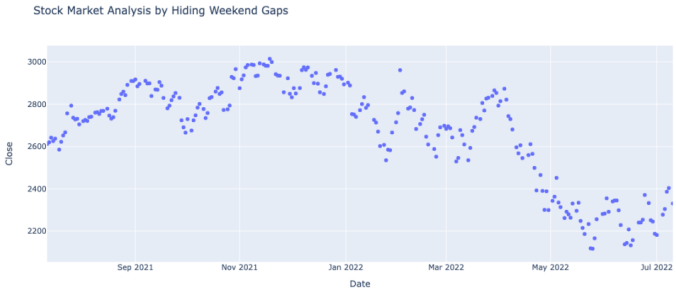
```
figure = px.line(data, x='Date', y='Close ,'
                 title='Stock Market Analysis with Time Period
Selectors('

figure.update_xaxes)
    rangeselector=dict)
        buttons=list])
            dict(count=1, label="1m", step="month",
stepmode="backward"),
```

```
            dict(count=6, label="6m", step="month",
stepmode="backward"),
            dict(count=3, label="3m", step="month",
stepmode="backward"),
            dict(count=1, label="1y", step="year",
stepmode="backward"),
            dict(step="all")
([
(
(
figure.show()
```



تؤثر عطلة نهاية الأسبوع أو موسم العطلات دائمًا على سوق الأوراق المالية. لذلك إذا كنت ترغب في إزالة جميع سجلات اتجاهات عطلة نهاية الأسبوع من تصوير سوق الأسهم الخاص بك، فيما يلي كيف يمكنك القيام بذلك:

```
figure = px.scatter(data, x='Date', y='Close', range_x=['2021-
07-12', '2022-07-11,['
                  title="Stock Market Analysis by Hiding
Weekend Gaps("
figure.update_xaxes)
    rangebreaks]=
        dict(bounds=["sat", "sun"])
[
(
figure.show()
```



هذه هي الطريقة التي يمكنك بها تحليل سوق الأسهم باستخدام بايثون. إذا كنت تريد معرفة كيفية التنبؤ بسوق الأوراق المالية، فيمكنك التعلم هنا.

الملخص

هذه هي الطريقة التي يمكنك بها استخدام لغة برمجة بايثون لتحليل سوق الأوراق المالية بشكل تفاعلي. يعني تحليل سوق الأسهم تحليل الاتجاهات الحالية والتاريخيةفي سوق الأوراق المالية لاتخاذ قرارات البيع والشراء المستقبلية. آمل أن تكون قد أحببت هذه المقالة حول تحليل سوق الأسهم باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2022/07/12/stock-market-analysis-using-python/

20) التنبؤ بالأعمال التجارية باستخدام بايثون Business Forecasting using Python

يعد التنبؤ بالأعمال التجارية (Business Forecasting) أحد تطبيقات التنبؤ بالسلاسل الزمنية (Time Series Forecasting).في تنبؤات الأعمال التجارية، نهدف إلى التنبؤ بالمبيعات أو النفقات أو الإيرادات المستقبلية باستخدام بيانات السلاسل الزمنية التاريخية التي تم إنشاؤها بواسطة الشركة. لذا، إذا كنت تريد معرفة كيفية إجراء التنبؤ بالأعمال التجارية، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بالأعمال باستخدام بايثون.

لماذا يحتاج العمل التجاري إلى التنبؤ بالأعمال التجارية؟

تبحث كل شركة عن استراتيجيات لتحسين أرباحها. يلعب متخصصو علم البيانات دورًا رئيسيًا في توفير أكثر التنبؤات دقةفي أي وقت. دائمًا ما تكون البيانات التي يتم إنشاؤها من قبل الشركة في متناول اليد لتحليل السلوك المستقبلي للعملاء المستهدفين. من خلال التنبؤ باتجاهات الأعمال المستقبلية، يمكن للشركة اتخاذ قرارات أفضل لتحسين أدائهافي المستقبل.

آمل أن تكون قد فهمت سبب احتياج الشركة اليوم إلى استخدام تقنيات التنبؤ بالأعمال. التنبؤ بالمبيعات أو الإيرادات أو النفقات هي بعض حالات استخدام التنبؤ بالأعمال التجارية. لذلك، في القسم أدناه، سوف آخذك خلال مهمة التنبؤ بالأعمال التجارية حيث سنهدف إلى توقع الإيرادات الفصلية لشركة Adidas. يتم جمع البيانات التي أستخدمها لهذه المهمة يدويًا من تقارير المبيعات ربع السنوية من Adidas. يمكنك تنزيل مجموعة البيانات من هنا.

التنبؤ بالأعمال التجارية باستخدام بايثون

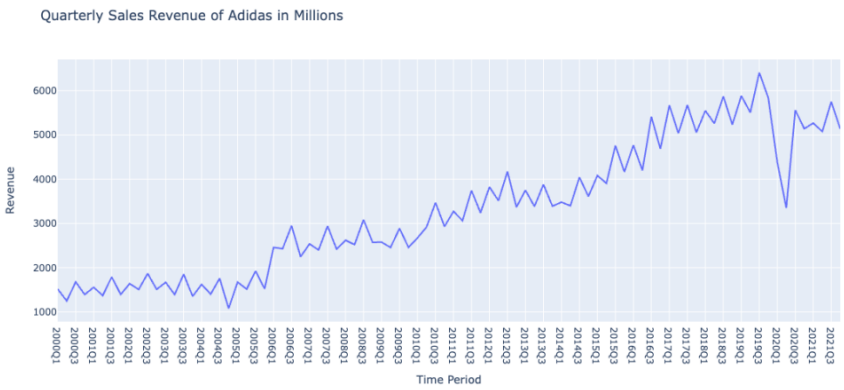
لنبدأ بمهمة التنبؤ بالأعمال التجارية عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
from datetime import date, timedelta
import datetime
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.arima_model import ARIMA
import statsmodels.api as sm
import warnings

data = pd.read_csv("adidas quarterly sales.csv")
print(data)
```

```
   Time Period  Revenue
0       2000Q1     1517
1       2000Q2     1248
2       2000Q3     1677
3       2000Q4     1393
4       2001Q1     1558
..         ...      ...
83      2020Q4     5142
84      2021Q1     5268
85      2021Q2     5077
86      2021Q3     5752
87      2021Q4     5137

[88 rows x 2 columns]
```

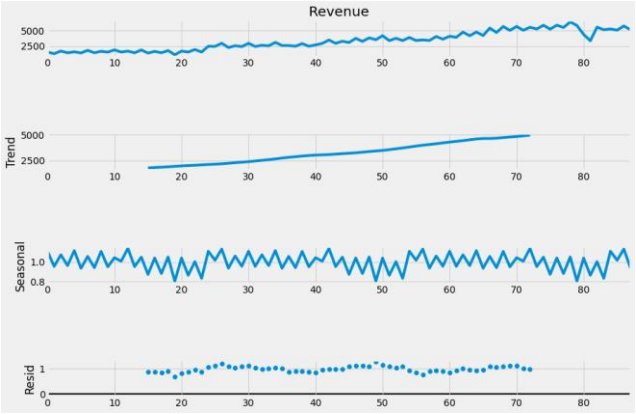
تحتوي مجموعة البيانات على عمودين؛ الفترة الزمنية (Time Period) والإيرادات (Revenue). يحتوي عمود الفترة الزمنية على الإيرادات ربع السنوية لشركة Adidas من 2000 إلى 2021، ويحتوي عمود الإيرادات على إيرادات المبيعات بالملايين (باليورو). دعونا نلقي نظرة على عائدات المبيعات ربع السنوية لشركة Adidas:

```
import plotly.express as px
figure = px.line(data, x="Time Period",
                 y="Revenue",
                 title='Quarterly Sales Revenue of Adidas in
Millions')
figure.show()
```



تعد بيانات إيرادات المبيعات الخاصة بشركة Adidas موسمية حيث تزيد الإيرادات الفصلية وتنخفض كل ربع سنة. فيما يلي كيفية التحقق من موسمية أي بيانات سلاسل زمنية:

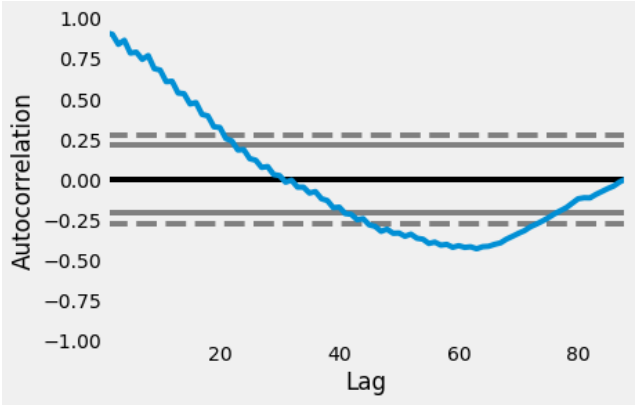
```
result = seasonal_decompose(data["Revenue ,["
                            model='multiplicative', freq = 30(
fig = plt.figure  ()
fig = result.plot  ()
fig.set_size_inches(10 ,15)
```



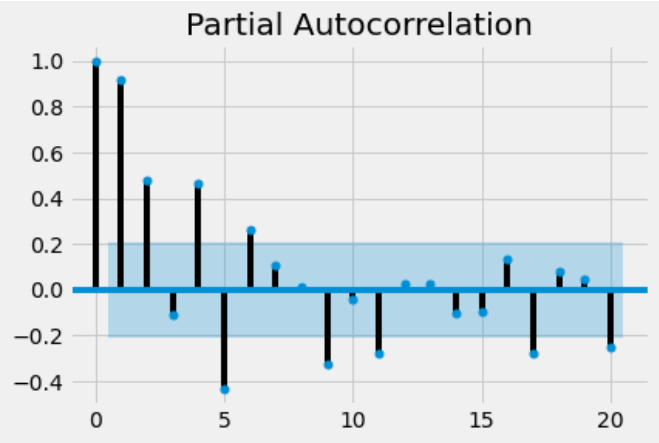
سأستخدم نموذج Seasonal ARIMA (SARIMA) للتنبؤ بإيرادات المبيعات ربع السنوية لشركة Adidas. قبل استخدام نموذج SARIMA، من الضروري إيجاد قيم p و d و q. يمكنك معرفة كيفية العثور على قيم p و d و q من هنا.

نظرًا لأن البيانات ليست ثابتة (not stationary)، فإن قيمة d هي 1. للعثور على قيم p و q، يمكننا استخدام مخططات الارتباط التلقائي (autocorrelation) والارتباط التلقائي الجزئي (partial autocorrelation plots):

```
pd.plotting.autocorrelation_plot(data["Revenue"])
```




```
plot_pacf(data["Revenue"], lags = 20)
```



الآن إليك كيفية تدريب نموذج SARIMA للتنبؤ بالإيرادات ربع السنوية لشركة Adidas:

```
model=sm.tsa.statespace.SARIMAX(data['Revenue"],
                                order=(p, d, q),
                                seasonal_order=(p, d, q, 12))
model=model.fit()
print(model.summary())
```

```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                           Revenue   No. Observations:                   88
Model:             SARIMAX(5, 1, 2)x(5, 1, 2, 12)  Log Likelihood                -548.520
Date:                            Mon, 05 Sep 2022  AIC                           1127.041
Time:                                    07:45:33  BIC                           1161.803
Sample:                                         0  HQIC                          1140.921
                                              - 88
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.5796      0.391     -4.044      0.000      -2.345      -0.814
ar.L2         -1.4321      0.587     -2.438      0.015      -2.583      -0.281
ar.L3         -0.8305      0.626     -1.328      0.184      -2.057       0.396
ar.L4         -0.5179      0.821     -0.630      0.528      -2.128       1.092
ar.L5         -0.2655      0.491     -0.541      0.589      -1.228       0.697
ma.L1          1.5056      0.518      2.906      0.004       0.490       2.521
ma.L2          0.9697      0.623      1.557      0.120      -0.251       2.190
ar.S.L12      -1.1270    362.141     -0.003      0.998    -710.910     708.656
ar.S.L24      -1.3418    312.728     -0.004      0.997    -614.277     611.594
ar.S.L36      -0.7832    174.955     -0.004      0.996    -343.688     342.122
ar.S.L48      -0.1847     50.633     -0.004      0.997     -99.423      99.054
ar.S.L60      -0.0098      8.921     -0.001      0.999     -17.496      17.476
ma.S.L12       0.3046    362.082      0.001      0.999    -709.363     709.972
ma.S.L24       0.8602    221.641      0.004      0.997    -433.548     435.269
sigma2      1.909e+05   4.01e+05      0.476      0.634   -5.96e+05    9.78e+05
==========================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):               427.98
Prob(Q):                              0.96   Prob(JB):                         0.00
Heteroskedasticity (H):               7.35   Skew:                            -2.04
Prob(H) (two-sided):                  0.00   Kurtosis:                        13.97
==========================================================================================
```

الآن دعونا نتوقع الإيرادات ربع السنوية لشركة Adidas للأرباع الثمانية القادمة:

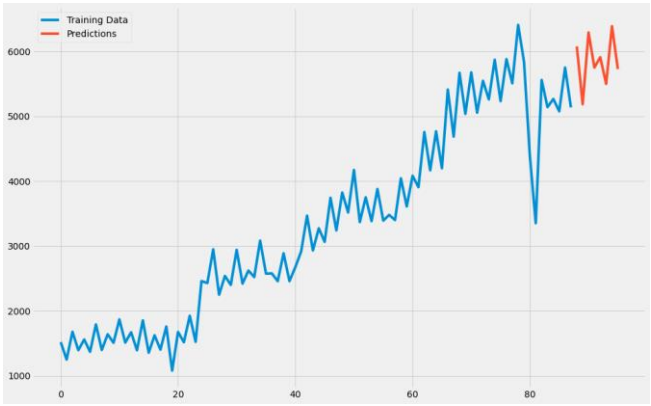
```
predictions = model.predict(len(data), len(data)+7)
```

```
print(predictions)
```

```
88    6078.793918
89    5186.311373
90    6293.196600
91    5751.905629
92    5911.946881
93    5499.784229
94    6389.627988
95    5728.806969
Name: predicted_mean, dtype: float64
```

إليك كيف يمكننا رسم التنبؤات:

```
data["Revenue"].plot(legend=True ,
                     label="Training Data ,"
                     figsize=(15, 10((
predictions.plot(legend=True, label="Predictions")
```



الملخص

هذه هي الطريقة التي يمكنك بها إجراء التنبؤ بالأعمال التجارية باستخدام لغة برمجة بايثون.في تنبؤات الأعمال التجارية، نهدف إلى التنبؤ بالمبيعات أو النفقات أو الإيرادات المستقبلية باستخدام بيانات السلاسل الزمنية التاريخية التي تم إنشاؤها بواسطة الشركة. أتمنى أن تكون قد أحببت هذه المقالة حول التنبؤ بالأعمال التجارية باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2022/09/05/business-forecasting-/using-python

21) 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.

المصدر:

https://thecleverprogrammer.com/2021/12/27/cryptocurrency-price-prediction-with-machine-learning/

22) 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 Covid-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()
```

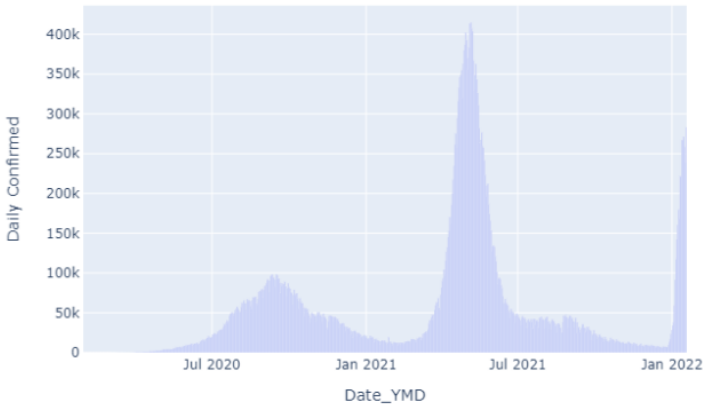
```
Date              0
Date_YMD          0
Daily Confirmed   0
Daily Deceased    0
dtype: int64
```

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()
```

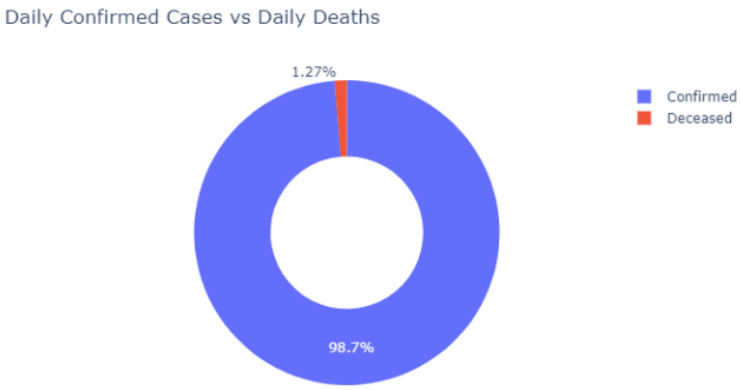


في تصوير البيانات أعلاه ، يمكننا أن نرى موجة عالية من حالات كوفيد ـ19 بين أبريل 2021 ومايو 2021.

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()
```



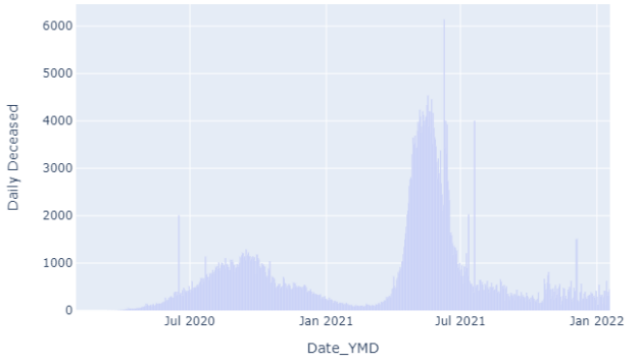
دعونا نحسب معدل الوفيات لـ Covid-19:

```
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.

المصدر:

https://thecleverprogrammer.com/2022/03/29/covid-19-deaths-prediction-with-machine-learning/

23) 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.

المصدر:

https://thecleverprogrammer.com/2021/10/22/tata-motors-stock-price-prediction-with-machine-learning/

24) 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.

المصدر:

https://thecleverprogrammer.com/2021/09/08/apple-stock-price-prediction-with-machine-learning/

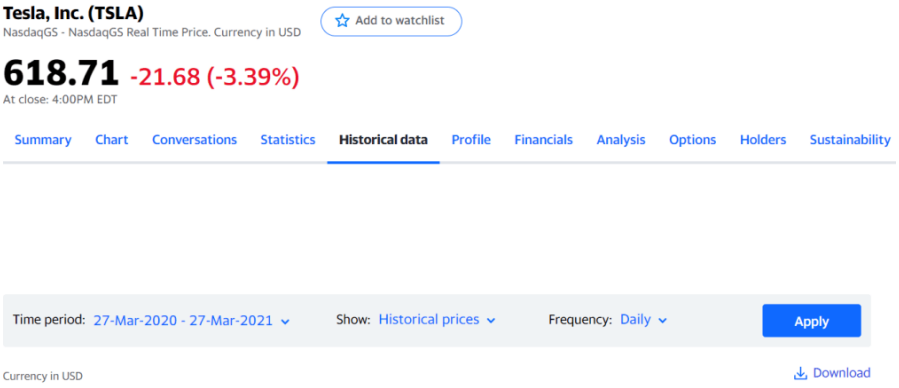
25) التنبؤ بسعر سهم Tesla مع التعلم الآلي Tesla Stock Price Prediction with Machine Learning

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 شخص في العالم. يُعد التنبؤ بأسعار الأسهم حالة استخدام رائعة للتعلم الآلي، لذلك في 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 Tesla في عيون العالم لفترة طويلة الآن حيث تدعم حكومات العديد من البلدان في جميع أنحاء العالم رؤية Tesla. لذلك في 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 موضح في 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 المذكورة في القسم أعلاه. دعنا الآن نرى كيفية توقع أسعار أسهم Tesla باستخدام التعلم الآلي باستخدام بايثون. سأبدأ هنا باستيراد مكتبات بايثون ومجموعة البيانات اللازمة:

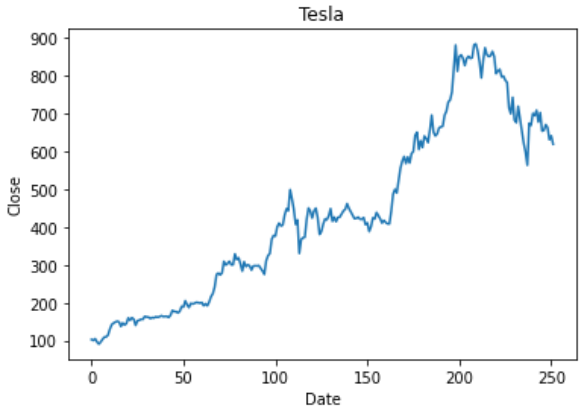
```
import numpy as np
```

```
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:

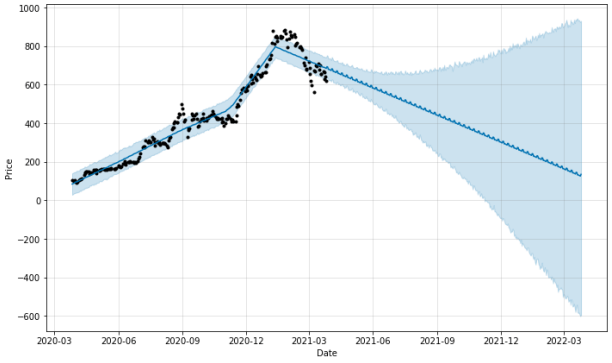
```
data = data.rename(columns={"Date" : "ds", "Close" : "y" })
```

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

المصدر:

https://thecleverprogrammer.com/2021/03/27/tesla-stock-price-/prediction-with-machine-learning

26) 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
```

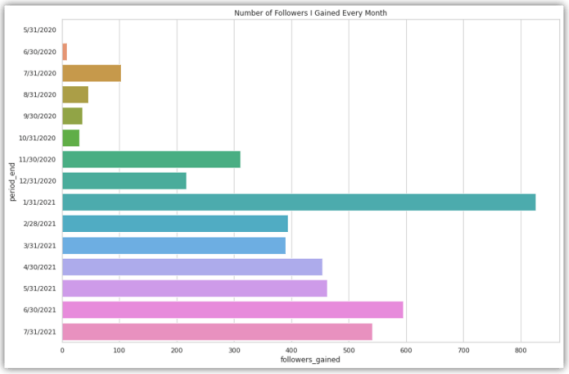
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

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]

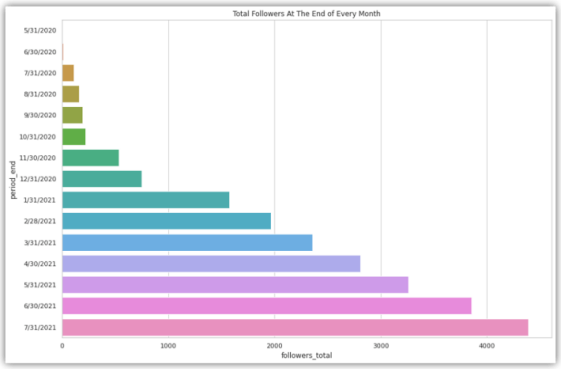
في السطر السابع من الكود أعلاه، قمت بحذف الصف الأخير من مجموعة البيانات لأنه يحتوي على بيانات حول هذا الشهر. الآن سألقي نظرة على عدد المتابعين الذين اكتسبتهم كل شهر على حسابي منذ أن انضممت إلى منصة التواصل الاجتماعي هذه:

```
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()
```



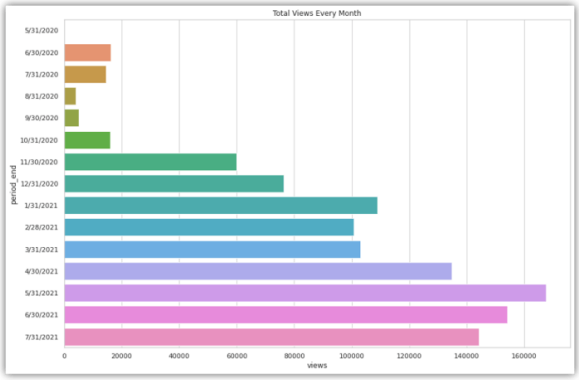
الآن دعنا نلقي نظرة على العدد الإجمالي للمتابعين الذين ينتهي بي الأمر معهم كل شهر:

```
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.

المصدر:

https://thecleverprogrammer.com/2021/08/09/social-media-followers-prediction-with-machine-learning/

27) 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 Elon Musk لشركة Tesla، 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 ".

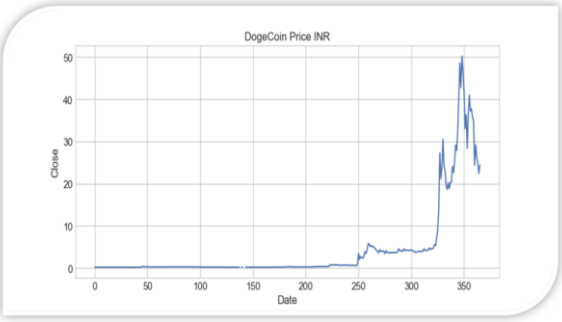
بعد الانتهاء من الخطوات المذكورة أعلاه، ستجد مجموعة بيانات بالأسعار التاريخية لـ Dogecoinفي مجلد التنزيلات الخاص بك. لنبدأ الآن بمهمة توقع أسعار Dogecoin عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
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.

المصدر:

https://thecleverprogrammer.com/2021/05/25/dogecoin-price-prediction-with-machine-learning/