Emma Taylor
Masters in financial transactions and business strategy
Lille, France.

Python for Finance
UBEYDULLAH Ozcan

PYTHON REPORT

Table of Contents

23rd of November 2023.

In order to implement a financial model using Python and creating it in a PyCharm environment I have chosen an easy financial model to be able to code effectively.

I- The model

The financial model used in this exercise is a forecasting model, using a straight line method. Meaning the expected years to come will be based on the previous revenue growth.

II- The data

Having to use an open source, accessible data, I decided to use the forecasting model on Apples revenue from the past 5 previous years. This data was accessed from Statista. Reference: Federica Laricchia (2023, Nov 6). *"Global revenue of Apple from 2004 to 2023".* Statista. https://www.statista.com/statistics/265125/total-net-sales-of-apple-since-2004/

Years	Apple's global Revenue (in billion's of dollars)
2018	265,60
2019	260,17
2020	274,52
2021	365,82
2022	394,33

III- The code

To understand step by step what the code is doing, I added the following commentary in the code:

Imports
Function
    # Reshape X to a column vector
    # Create and fit the linear regression model
    # Generate future X values for forecasting
    # Predict future y values
Apple's data from Statista
Number of periods to forecast
Plot of the original data and the forecast

Note that the values representing the years named 'time in years' and the revenue named 'value in billion of dollars' in the code have been named in this report as 'time' and as 'value' respectively for simplification.

Imports

In this program I imported pandas and numpy libraries.

- Pandas (pd) was used for its data structures, such as 'DataFrame' for the two-dimensional table, with 'time' and 'value' representing the years and Apples' global Revenue respectively.
- NumPy (np) was used here to support the multinational-array, to perform numerical operations for creating a range of values for future time periods. The np.arrange and np.reshape are used to generate an array with evenly spaced values to reshape arrays respectively.

The function

To make our forecasting we used a linear regression model using the 'sklearn.linear_model'. I used this model as I know it is a statistical method used to model the relationship between a dependent variable, here apple's global revenue, and one or more independent variables, here the years. The 'sklearn' from 'scikit-learn' provides the code with consistent and easy to use API. It comes with a simple and convenient API that makes it easier to create, train and use the linear regression model.

I defined the forecast-straight-line method function to enclose the logic related to the forecasting values using a straight-line method. It also enabled the creation of the linear regression model, generating future 'X' values (the future years) and predicting future 'y' values (future revenue values).
The function takes the parameters ('X', 'y', and 'future_periods').

The parameter 'X' represents the independent variable; the year, named 'time in years' in the code.
Reshaping X with (-1, 1) was key to ensure that each row represents a sample, and each column represents a feature, as the linear regression expects the input feature of 'X' to be a 2D array. The input is reshaped for 'X' to be reshaped as a column vector.
Here future_X represents the future values of the independent variable (the years: 'time') for which the linear regression model is making predictions. These values are generated based on the last observed time value in the original dataset ('X'), and additional future periods specified by the future_periods variable.

The parameter 'y' represents the dependent variables; Apple's revenue, named 'value in billions of dollars' in the code.
Similarly, the variable future_periods represent the number of future periods for which the linear regression model is making predictions. It is used to determine how many future time points to generate for the forecast. To this variable I assigned the number 3 as I expect the forecast to show apple's forcasted revenue for the next 3 years.

To understand deeper, I have defined the other variables:

- X.max(): This finds the maximum value in the original time series X, representing the last observed time. Here X.max is 2022, as we have recorded revenue until 2022.
- np.arange(X.max() + 1, X.max() + 1 + future_periods): This creates an array of future time values starting from one more than the last observed time up to future_periods

more periods (here 3). The + 1 is used to start the sequence from the next time point.

What we really expect from the model is shown in the variable future_y. It represents the forecasted values for the future time points generated by the linear regression model. These are the values that the model estimates based on the learned relationship between the time variable and the dependent variable (in this case, the 'Value' variable).
As we are using a linear regression for the straight-line method, the variable is based on the previous revenue growth of Apple, the revenue growth between 2021 and 2022.

Formatting the output

As we are seeking a forecast, I thought it was easier to see the results through a scatter plot, to read where the previous revenue values were and what would the forecasted revenue looks like. To obtain these results, the 'matplotlib.pyplot', added in my environment, helped create visualizations, specifically to link to original data points and their forecasted values. Matplotlib allowed me to create static, animated, and interactive visualizations. Pyplot provided a MATLAB-style interface to create the plot.

These next 3 codes were specified for the design of the plot:

- plt.scatter(data['Time'], data['Value'], label='Original Data'): This line creates a scatter plot of the original data points with time on the x-axis and values on the y-axis.

- plt.plot(future_X, future_y, label='Forecast', color='red', linestyle='dashed'): This line creates a line plot (dashed line) for the forecasted values. The forecasted values are plotted in red to mark the difference between the previous and forecasted revenue values.

- The subsequent lines of code (plt.xlabel('Time'), plt.ylabel('Value'), plt.legend(), and plt.show()) are used to add labels to the axes, display a legend, and show the plot.

Results

As a result we obtain a scatter plot graph, where we can see Apple's revenue from the 5 previous years. From these results we can also see how Apple really knew how to come back from COVID-19's sales results, where between 2020 and 2021, their sales nearly increased by a 100 billion dollars.
From 2023 to 2025 we can observe a straight-line forecast of Apple's revenue growth, where for each year, the revenue growth is increased, based on the year before, starting from 2021/2022 as that is the last revenue growth recorded.