BUSINESS FORECASTING

What is Business Forecasting?  
  
Business Forecasting is one of the applications of Time Series Forecasting. In Business Forecasting, we aim to forecast future sales, expenditure, or revenue by using the historical Time Series data generated by the business.

**Why does a Business Needs Business Forecasting?**

Every business is looking for strategies to improve its profits. Data science professionals play a major role in providing the most accurate predictions for any given time. The data generated by a company is always handy for analyzing the future behaviour of target customers. By predicting future business trends, a business can make better decisions to improve its future performance.

Forecasting sales, revenue or expenditure are some use cases of business forecasting. So, business forecasting where we will aim to predict the quarterly revenue of Adidas. The data I am using for this task is collected manually from quarterly sales reports from Adidas.

import pandas as pd

2

from datetime import date, timedelta

3

import datetime

4

import matplotlib.pyplot as plt

5

plt.style.use('fivethirtyeight')

6

from statsmodels.tsa.seasonal import seasonal\_decompose

7

from statsmodels.graphics.tsaplots import plot\_pacf

8

from statsmodels.tsa.arima\_model import ARIMA

9

import statsmodels.api as sm

10

import warnings

11

​

12

data = pd.read\_csv("adidas quarterly sales.csv")

13

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]**

The dataset contains two columns; Time Period and Revenue. The Time Period column contains the quarterly revenue of Adidas from 2000 to 2021, and the Revenue column contains the sales revenue in millions (euros). Let’s have a look at the quarterly sales revenue of Adidas:

1

import plotly.express as px

2

figure = px.line(data, x="Time Period",

3

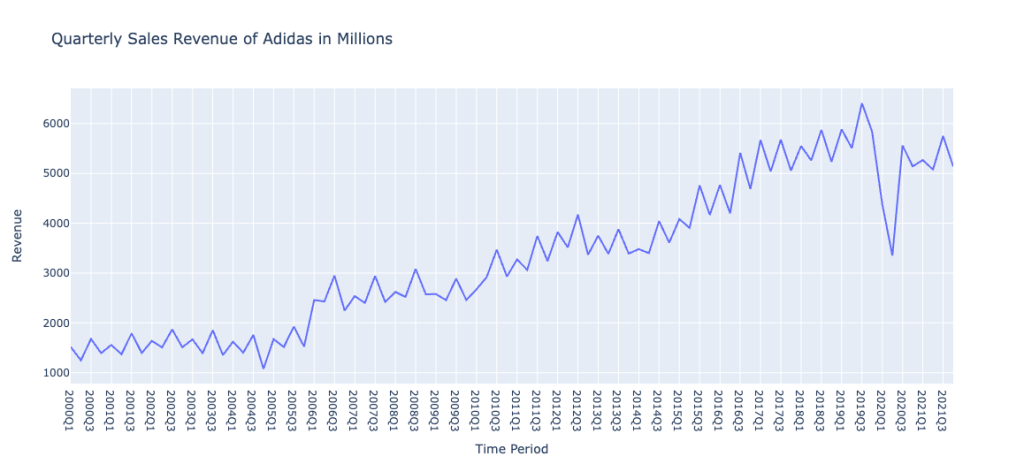
y="Revenue",

4

title='Quarterly Sales Revenue of Adidas in Millions')

5

figure.show()



The sales revenue data of Adidas is seasonal as the quarterly revenue increases and decreases every quarter. Below is how we can check the seasonality of any time series data:

1

result = seasonal\_decompose(data["Revenue"],

2

model='multiplicative', freq = 30)

3

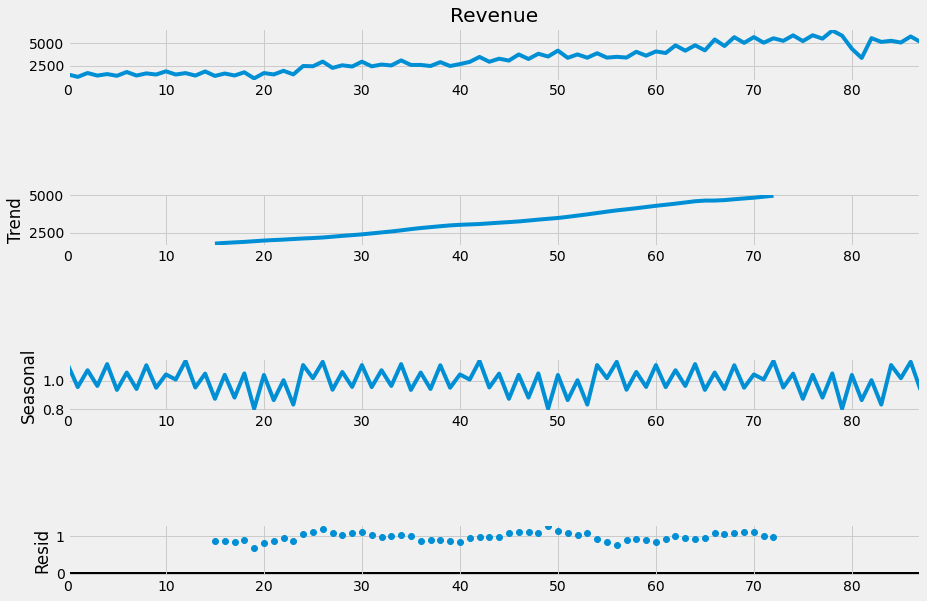
fig = plt.figure()

4

fig = result.plot()

5

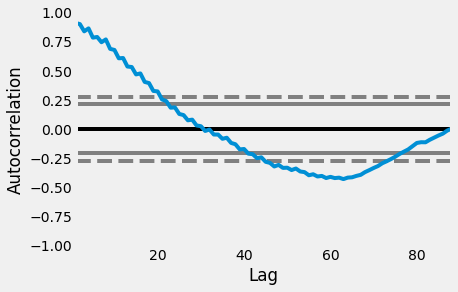
fig.set\_size\_inches(15, 10)



I will use the Seasonal ARIMA (SARIMA) model to forecast the quarterly sales revenue of Adidas. Before using the SARIMA model, it is necessary to find the p, d, and q values.

As the data is not stationary, the value of d is 1. To find the values of p and q, we can use the autocorrelation and partial autocorrelation plots:

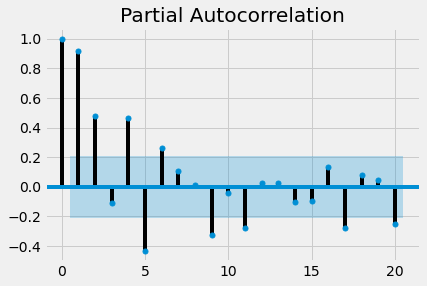
1

pd.plotting.autocorrelation\_plot(data["Revenue"])

***P Value = 5***

1

plot\_pacf(data["Revenue"], lags = 20)

***Q Value = 2***

Now here’s how to train a SARIMA model to predict the quarterly revenue of Adidas:

1

model=sm.tsa.statespace.SARIMAX(data['Revenue'],

2

order=(p, d, q),

3

seasonal\_order=(p, d, q, 12))

4

model=model.fit()

5

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**

**===================================================================================**

Now let’s forecast the quarterly revenue of Adidas for the next eight quarters:

1

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

2

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**

Here’s how we can plot the predictions:

1

data["Revenue"].plot(legend=True,

2

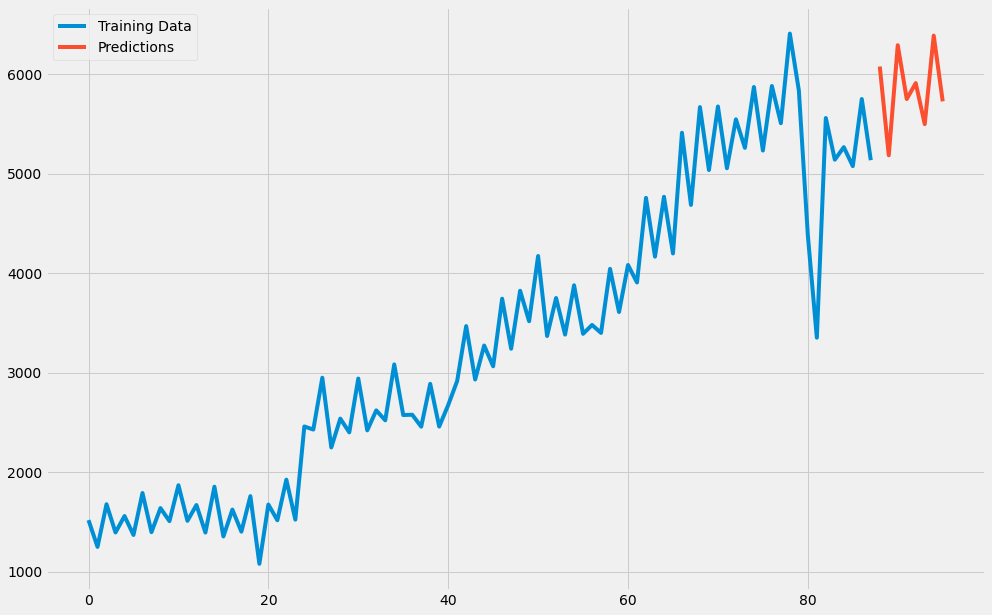
label="Training Data",

3

figsize=(15, 10))

4

predictions.plot(legend=True, label="Predictions")



**Summary**

So this is how you can perform business forecasting using the Python programming language. In Business Forecasting, we aim to forecast future sales, expenditure, or revenue by using the historical Time Series data generated by the business. I hope you liked this article on Business Forecasting using Python.

PROJECT REPORT   
BY TEAM   
Alfred Nirmal Raj A  
Prakash R  
Akash N