**D213 Performance Assessment Task 1**

Renzo Espinoza

College of Information Technology, Western Governors University

D213: Advanced Data Analytics

Dr. Festus Elleh

March 3, 2023

**A1:Research Question**

A question relevant to a real-world organizational situation that I will answer by using time series modeling techniques is: can I accurately predict the daily revenue of the company for the next quarter?

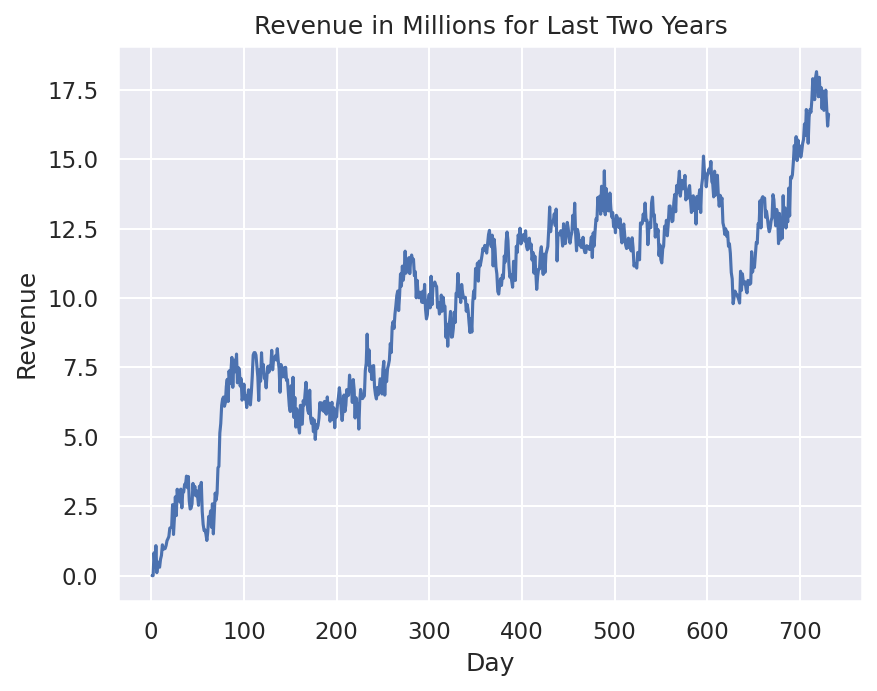
**A2:Objectives and Goals**

The goal of my analysis is to not only forecast future revenue, but to also identify trends in the data. Identifying patterns in the time series data can help the company determine whether there is a link between customer churn and revenue.

**B:Summary of Assumptions**

The forecasting model that I will be using, autoregressive integrated moving average model (ARIMA), assumes the time series is stationary. According to Peixeiro (2022), a time series is stationary when it has a constant mean, variance, and autocorrelation. Peixeiro defines autocorrelation as the linear relationship between a time series and a lagged version of itself.

**C1:Line Graph Visualization**



**C2:Time Step Formatting**

The time step formatting of the realization is a day, the length of the time series is 731 days. According to the line graph, there are no missing gaps in measurement of the data.

**C3:Stationarity**

To evaluate the stationarity of the time series, I used the augmented Dickey-Fuller test. The Statsmodels library contains a module called ‘adfuller’ that allows me to perform the test. The ADF test returned a p-value of 0.32 and a test statistic of -1.92, because the ADF statistic is a small negative number and the p-values is much greater than 0.05, we cannot reject the null hypothesis that the time series is not stationary. The data will need to be transformed for it to be properly processed by the ARIMA model.

**C4:Steps to Prepare the Data**

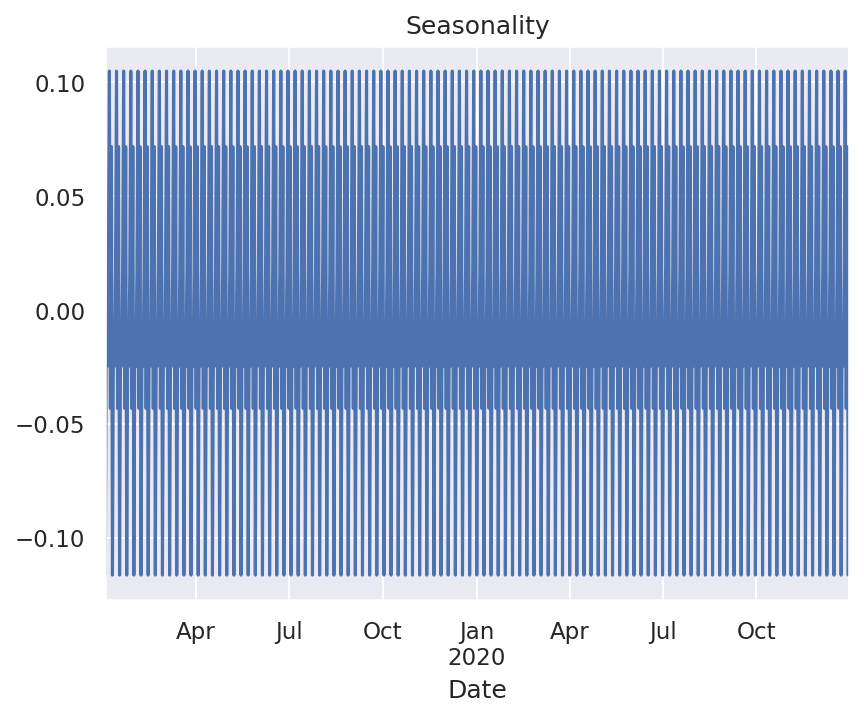
The very first step to prepare the data analysis was to import the raw CSV file into a data structure so I could view its contents and manipulate it. The Pandas library provides that functionality with its ‘read\_csv’ method and ‘Dataframe’ data srtucture. Using Matplotlib and Seaborn, I visualize the data to check for missing or repeated values. Thankfully, no input errors appeared in the dataset, the only issue is that the time series is not stationary. I then performed differencing on the ‘Revenue’ column using the ‘diff’ method from the Numpy library, this successfully transformed the data to become stationary. Certain library functionionality require the time series data to have dates as their index, so I used the Pandas library to replace the index of each entry in the data with dates starting from January 1st, 2019, spanning across two years. Finally, I split the data into training and test sets by index slicing the ‘Dataframe’ into a 75/25 train/test split.

**C5:Prepared Dataset**

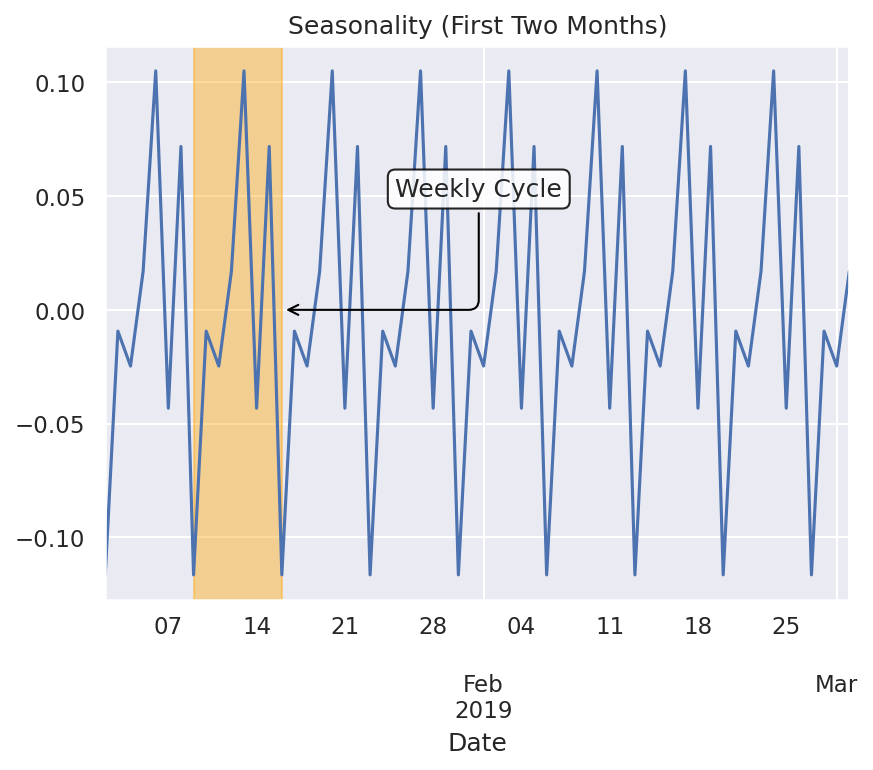
The cleaned dataset will be attached to my submission as ‘PA1\_cleaned\_data.csv’.

**D1:Report Findings and Visualizations**

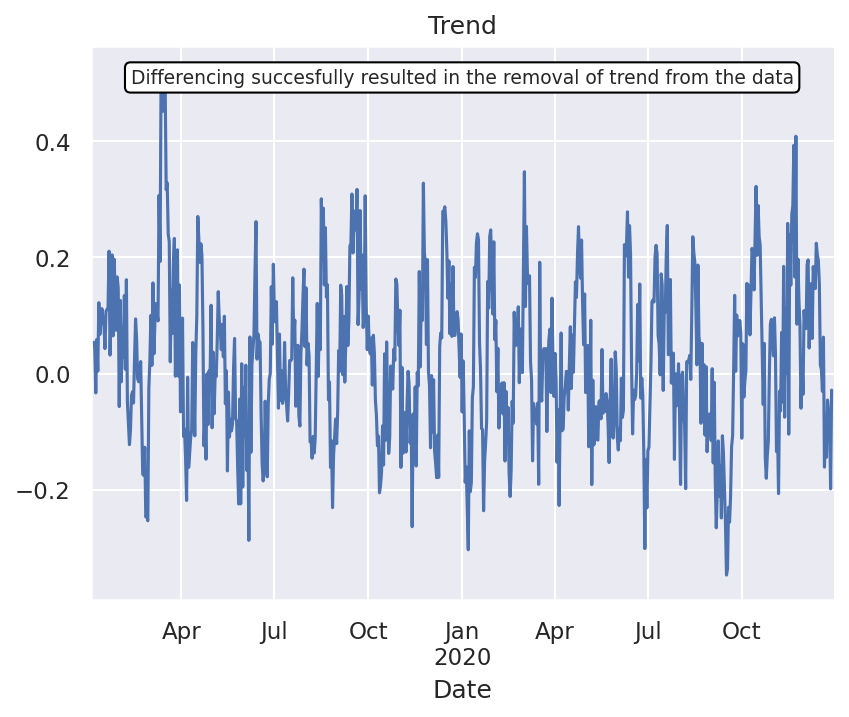
Here is a plot of the seasonality of the differenced time series:

****

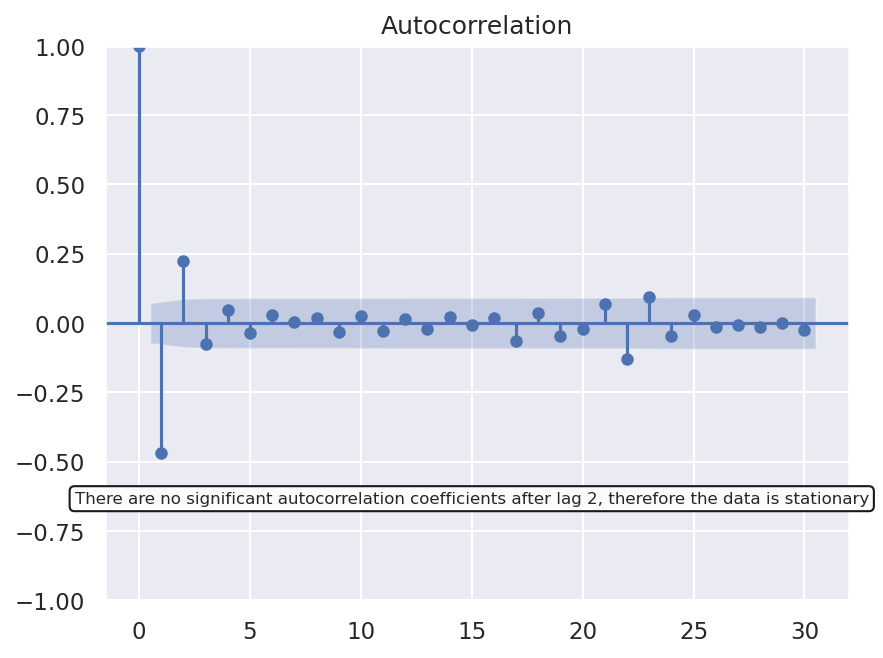
Seasonality is clearly present in the data, as there is a repeating pattern at regular intervals. To understand the length of this cycle we will take a closer look. After visualizing the first couple months of data, it becomes clear that the regular fluctuations occur on a weekly seasonal cycle. Therefore, the period for seasonal differencing, m, will be set to 7 for the ARIMA model.

****

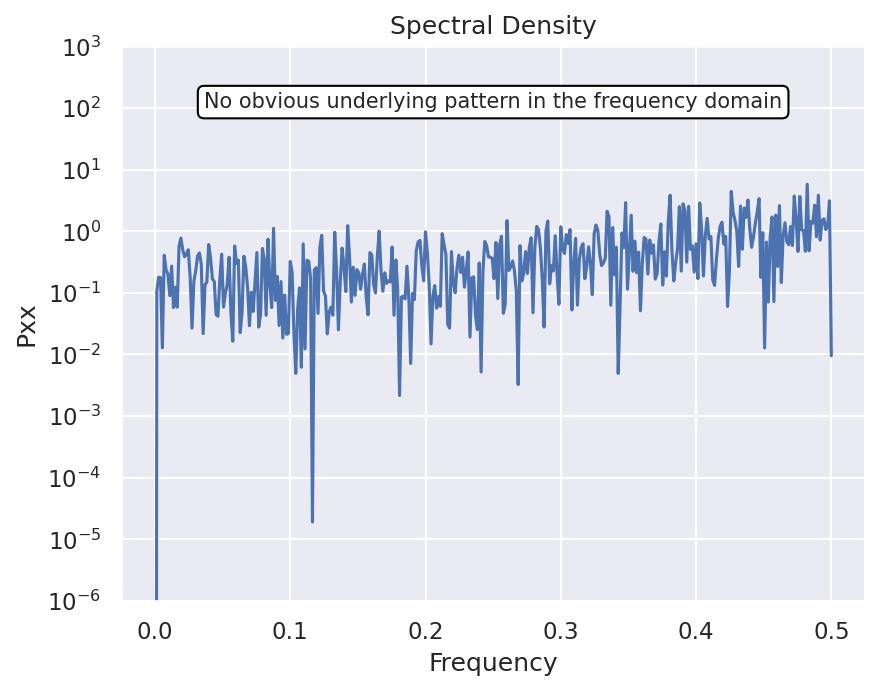
The trend of the time series is shown below. Differencing the data successfully removed the upward trend that was apparent in the original time series.

****

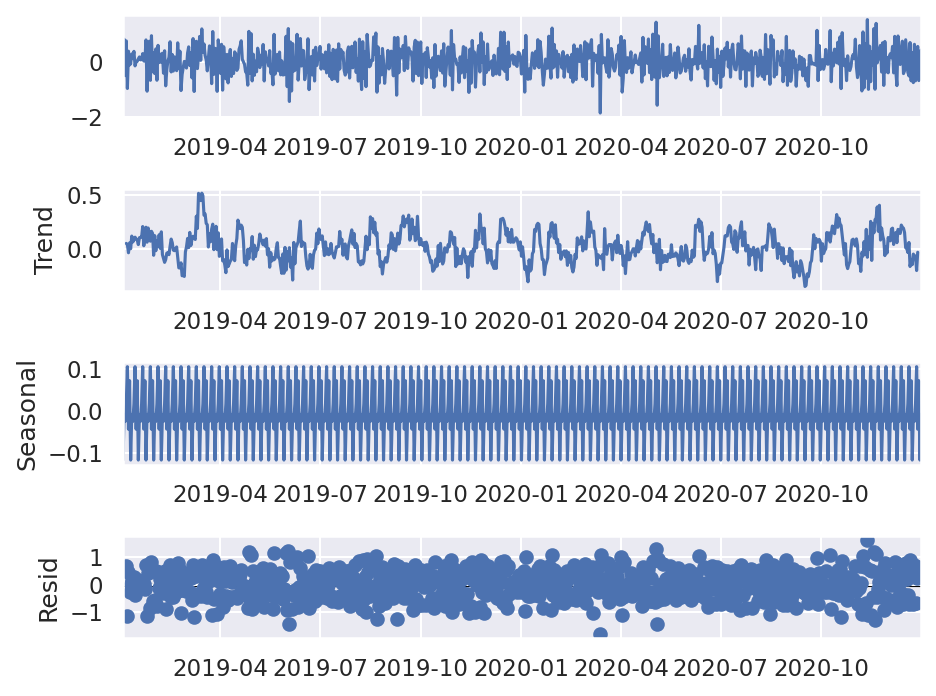
Using the ‘plot\_acf’ function from the Statsmodels library, I plotted the autocorrelation of the cleaned data. As you can see, after lag-2 there are no significant autocorrelation coefficients. This further confirms that diferencing the data by one lag successfully coerced stationarity.

****

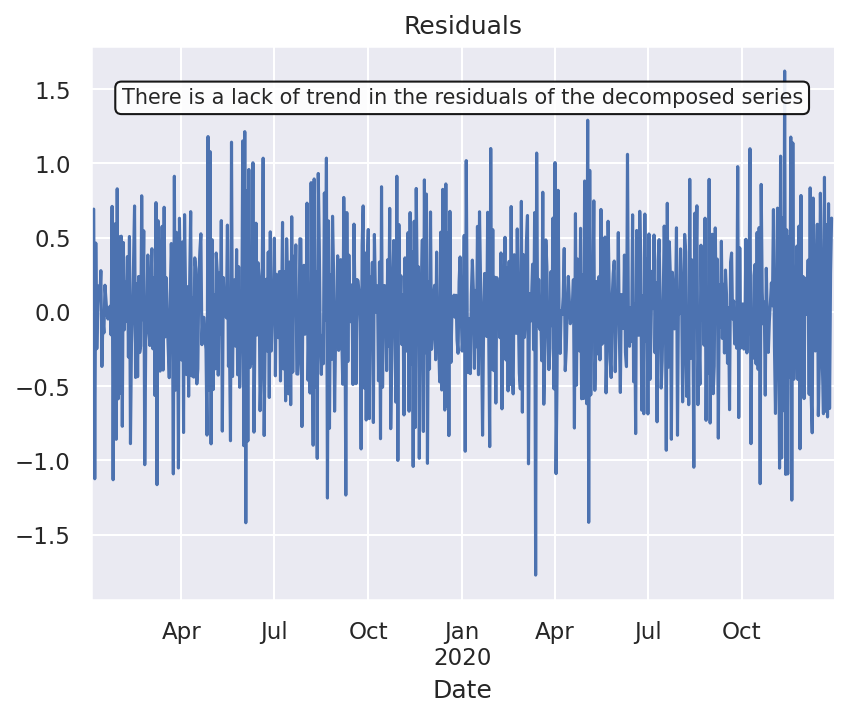
Below is a plot of the power spectral density using a periodogram, this functionality came from the SciPy library. No obvious underlying pattern appears in the frequency domain.

****

The decomposed time series plot provided by the Statsmodels library is displayed below. This breaks down the cleaned time series into its three components: seasonality, trend, and residuals.

****

Let’s take a clearer look at the residuals. The Statsmodels library provides a function to plot the residuals of the time series. Using this function, it appears that there is a lack of trend in the residuals of the data, resembling white noise.

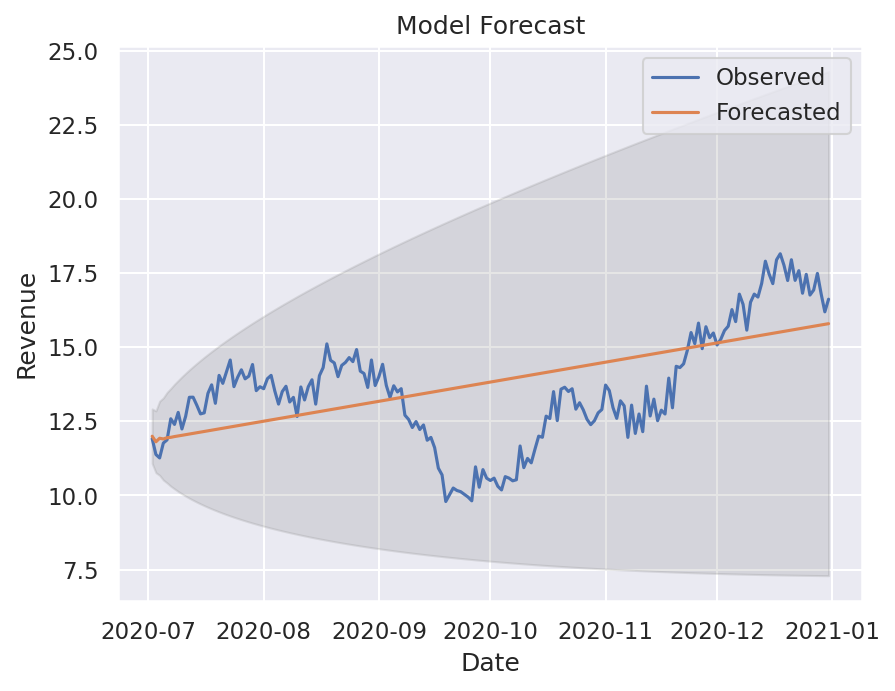


**D2:ARIMA Model**

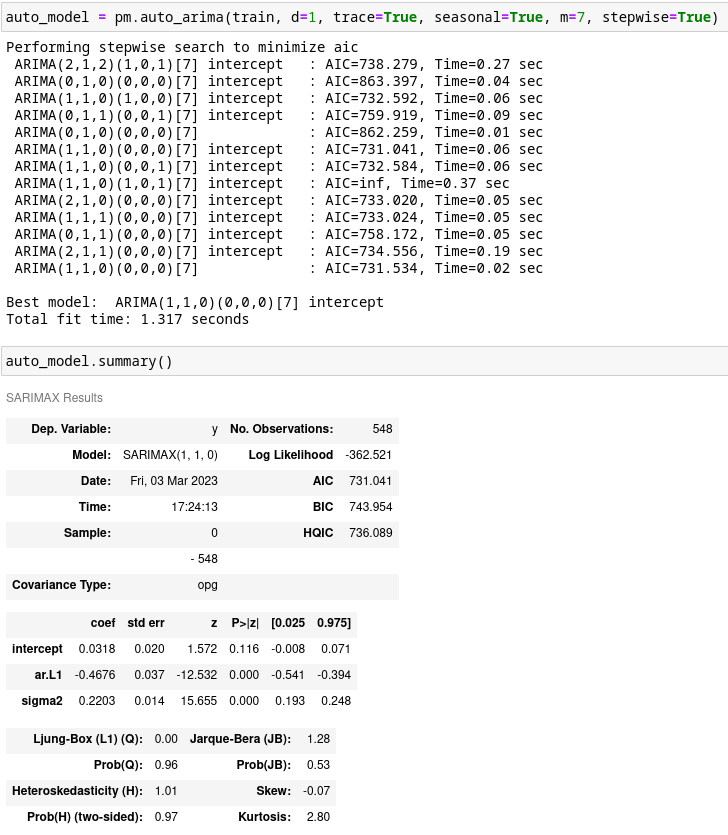
To identify an ARIMA model that takes into account the observed trend and seasonality of the data, I used the Pmdarima library and its ‘auto\_arima’ function to search through different combinations of *p* and *q*, as well as different combinations of *P, D,* and *Q* because there is seasonality present in the time series. The combination of parameters that has the lowest AIC score is chosen. The value of  *d* remains constant at 1, because it took one round of differencing for the data to become stationary. According to Peixeiro (2022), the frequency parameter, *m*, should be set to 7 for daily data that has a weekly seasonality. The search for the optimal seasonal ARIMA model resulted in a model with the parameters 1, 1, and 0 for *p, d,* and *q* respectively. The optimal model according to the ‘auto\_arima’ method selection proccess contains no seasonal component, in other words *P, D,* and *Q* are 0.

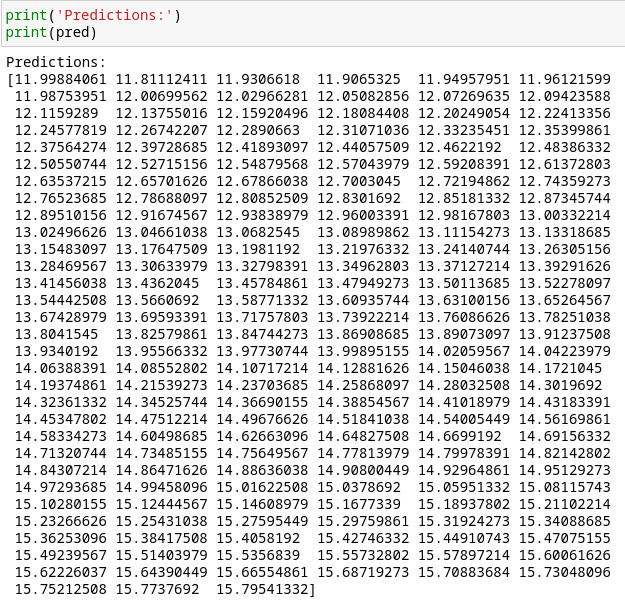
**D3:Forecasting using ARIMA Model**

With the ‘predict’ method from the Pmdarima library, I forecasted the next 6 months of data for the training set and then compared the results with the test set.

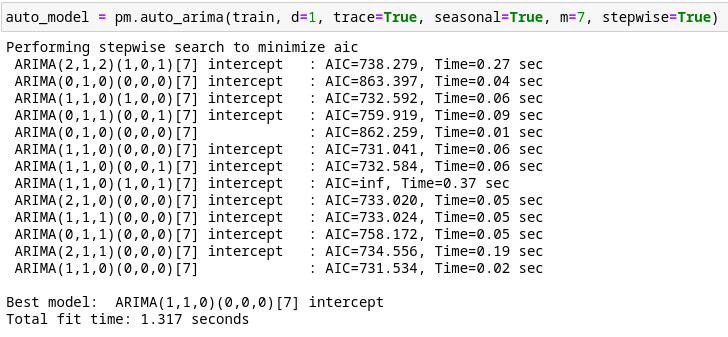
****

**D4:Output and Calculations**

****

****

**D5:Code**

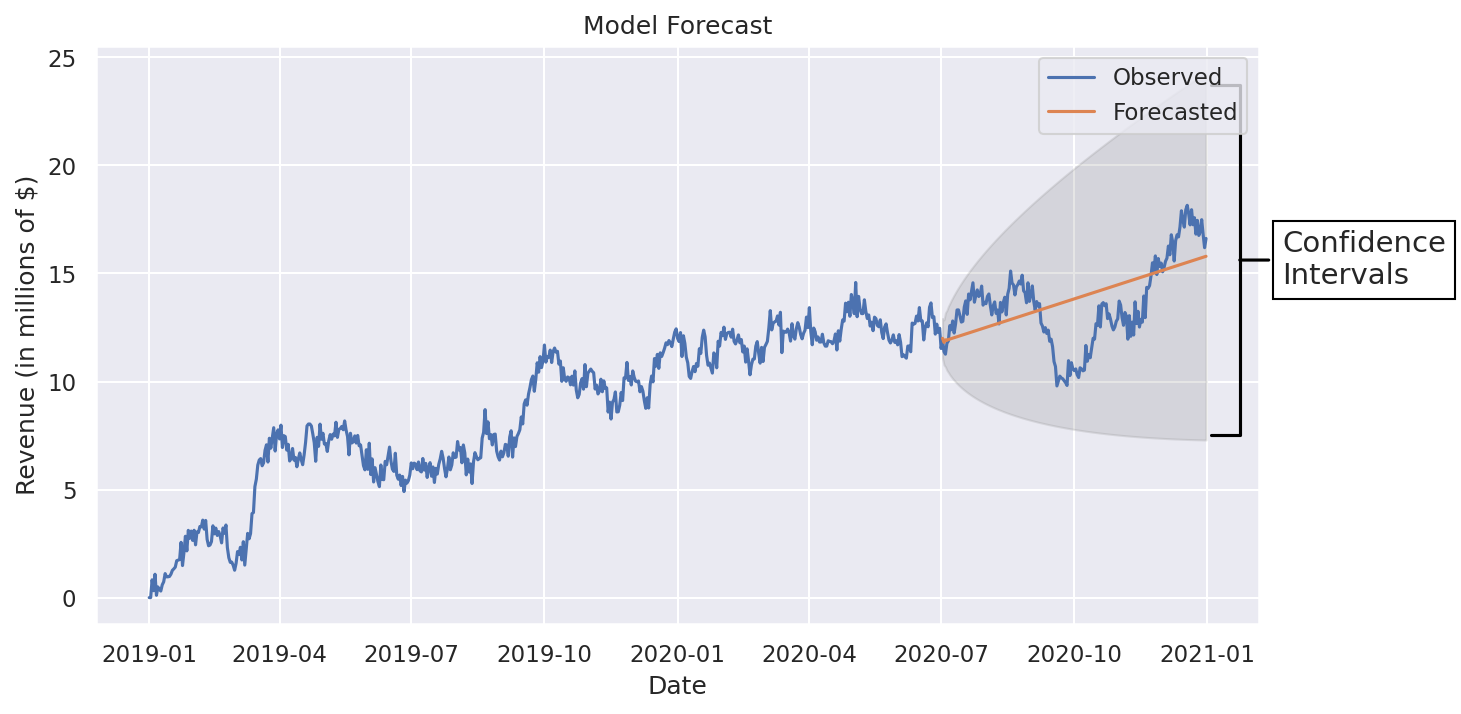
****

**E1:Results**

The selection of the ARIMA model was implemented by the ‘auto\_arima’ function from the Pmdarima library. The model evaluation procedure works by trying several different combinations of model parameters, and then choosing one of the models based on a model evaluation metric, I chose AIC. According to Pmdarima’s documentation for the auto\_arima function, performing a grid search over every possible combination of parameters is especially slow for seasonal data, so instead I used the “stepwise” algorithm approach for testing hyperparameter combinations. The ARIMA model with its parameters set to (1,1,0) (0,0,0) had the lowest AIC at 731.534.

The prediction interval of the forecast is one dayand the length of the forecast is around 6 months. I chose 6 months as the length of forecast because 75/25 is a common training/test split used in time series forecasting, so a reliable 6 month forecast should be possible with over 18 months of training data. The root mean squared error between the test set and the predicted set is about 1.78. The RMSE is slightly high considering the data ranges from 0 to 16 and the unit of measurement is in millions of dollars. I calculated the mean absolute percentage error to get a scale-independent error metric. It resulted in a MAPE of 11.83%, which is a good score according to Allwright (2022).

**E2:Annotated Visualization**

****

**E3:Recommendations**

Overall, the model does a good job of capturing the trend of the data and has a good MAPE and RMSE score. I would recommend using this model to project the revenue for the next two quarters. There is of course always room for improvement, I also recommend testing more combinations of model hyperparameters than I did for my analysis. Including seasonality into the ARIMA model could be promising, but from my testing, introducing seasonality lead to a more complex but far less accurate model.

**F:Reporting**

My report will be attached to my submission titled ‘PA1 Report.html’.

**G & H:Sources**

Allwright, S. (2022, December 6). What is a good MAPE score? (simply explained). Stephen Allwright. Retrieved March 4, 2023, from https://stephenallwright.com/good-mape-score/

Peixeiro, M. (2022). *Time series forecasting in Python*. Manning Publications Co.

*pmdarima.arima.auto\_arima*. pmdarima 2.0.2 documentation. (n.d.). Retrieved March 4, 2023, from https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto\_arima.html