Gaby Masak

D603 – Machine Learning

Task 3: Time Series Modeling

11/25/2024

Explanation: Code for Data Production Pipeline

**Requirement A: Gitlab Subgroup and Project**

GitLab URL: <https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d603-machine-learning/-/blob/working_branch/D603Task3.py?ref_type=heads>

Screenshot of Repository Branch History:

A screenshot of a computer

Description automatically generated

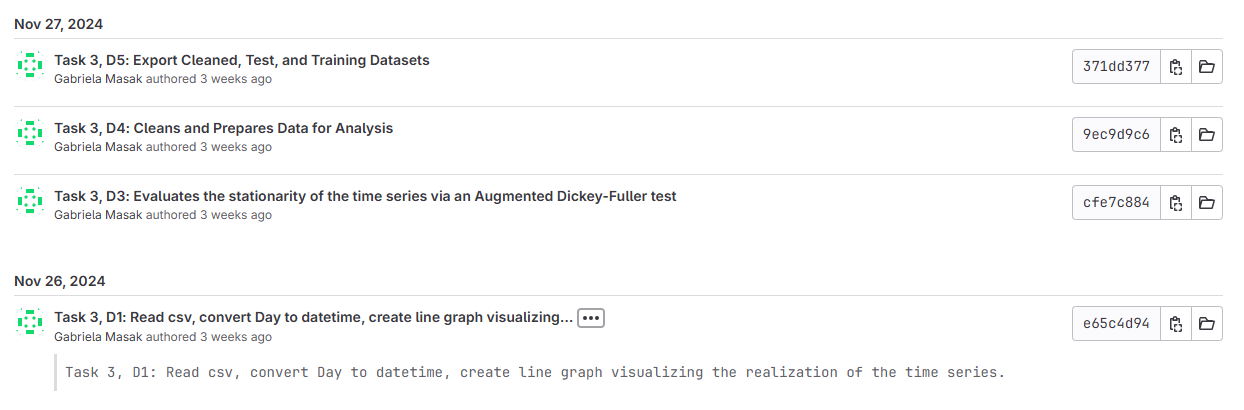
****

Figure 1: Screenshots of Repository Branch History

Please Note: D1 and D2 were initially submitted together, noted with “convert Day to datetime” under the November 26, 2024 commit. Another commit was created to separate for the sake of the submission requirements.

**Requirement B: Purpose of Data Analysis**

Hospitals are frequently penalized and fined by the Centers for Medicare and Medicaid Services (CMS) for excessive readmission. Motivated to decrease readmissions, hospitals implement various projects. In order to answer questions about the impact of these projects on revenue, time series modeling techniques can be employed. Analyzing with time series modeling can answer research questions about how specific interventions aimed at reducing hospital readmissions impact overall hospital revenue over time as well as both whether and how these interventions will continue to affect hospital revenue in the subsequent financial quarter. The objectives of these types of data analysis revolve around uncovering trends and patterns in revenue over time. By analyzing these trends, hospitals can identify fluctuations in revenue and correlate them with specific interventions or changes in hospital practices. This analysis can help determine the effectiveness of implemented projects in reducing readmissions and improving patient outcomes. Additionally, understanding the financial impact of these projects on hospital revenue can guide future decision-making and resource allocation, ensuring that efforts to reduce readmissions are both clinically effective and economically sustainable.

**Requirement C: Assumptions of Time Series Model (Stationary, Autocorrelated Data)**

Autocorrelation and stationarity are fundamental concepts in time series modeling that directly relate to the assumptions and data cleaning and model selection process. Stationarity refers to the property of a time series where its statistical properties, such as mean, variance, and autocorrelation, remain constant over time. Types of stationarity include trend, seasonal, and strict stationary (Kumar, 2021). Many time series models, including AutoRegressive Integrated Moving Average (ARIMA), assume that the data is stationary because non-stationary data can lead to unreliable and inaccurate forecasts. To achieve stationarity, data preprocessing steps such as differencing, detrending, and seasonal adjustment are often applied. These steps help to stabilize the mean and variance, making the time series suitable for modeling and ensuring that the underlying data-generating process does not change over time.

Autocorrelation, on the other hand, measures the correlation between observations in a time series at different time lags. High autocorrelation can indicate that past values significantly influence current values. Understanding the autocorrelation structure is crucial for selecting appropriate model parameters and improving forecast accuracy (Muralidhar, 2021). During the data cleaning process, identifying and addressing autocorrelation helps in ensuring that the residuals (errors) from the model are uncorrelated, which is a key assumption for many time series models.

Time series models rely on several key assumptions to produce accurate forecasts: stationarity, independence of errors, and homoscedasticity. With reference to autocorrelation, the assumption of independence of errors states that the residuals (errors) from the model should be uncorrelated with each other. In other words, the error terms should not exhibit any patterns or correlations. Violations of this assumption can lead to biased parameter estimates and incorrect inferences. Additionally, homoscedasticity, which requires that the variance of the residuals remains constant over time, is crucial for ensuring that the model's predictions are equally reliable across different time periods. If the variance of the residuals changes over time, thereby exhibiting heteroscedasticity, it can affect the model's accuracy and reliability.

**Requirement D: Data Cleaning and Preparation**

Several steps were taken to prepare the data for analysis. First, a line graph was plotted using Python to visualize the realization of the time series, revenue over time, seen in Figure 2.

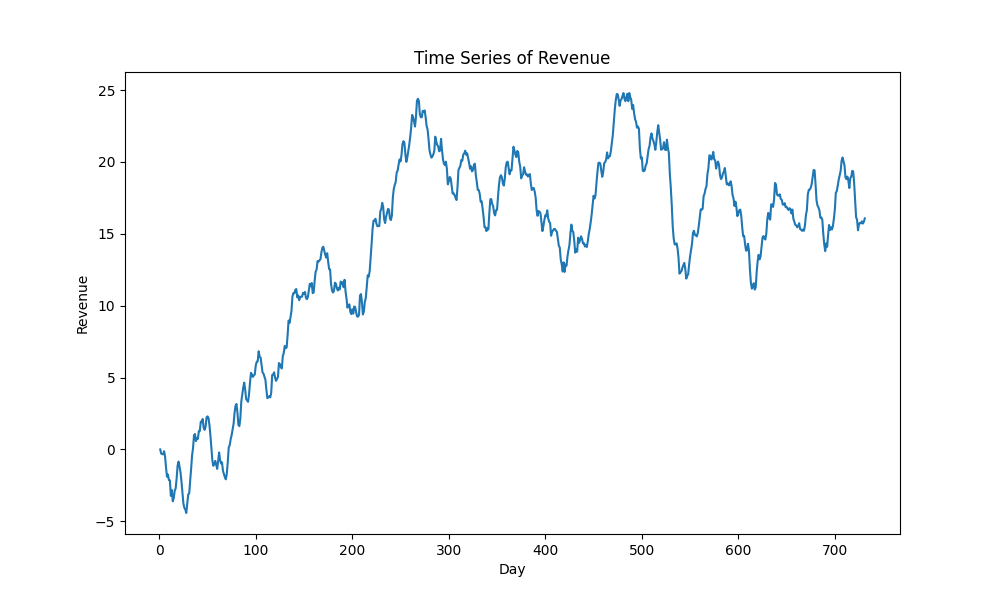


Figure 2: Line Graph of Revenue Time Series

Next the day column in the data frame indicated formatting with daily time steps, 371 total, or a period of approximately 2 years. The data dictionary does not indicate exactly when this period starts or ends, instead referencing the start of an ambiguous program surrounding patient readmittance. As a result, the script defines the first day of the column as January 1, 2021 as an estimation, converting the subsequent data to datetime formatting. This modification is reflected in Figure 3 below along the x-axis.

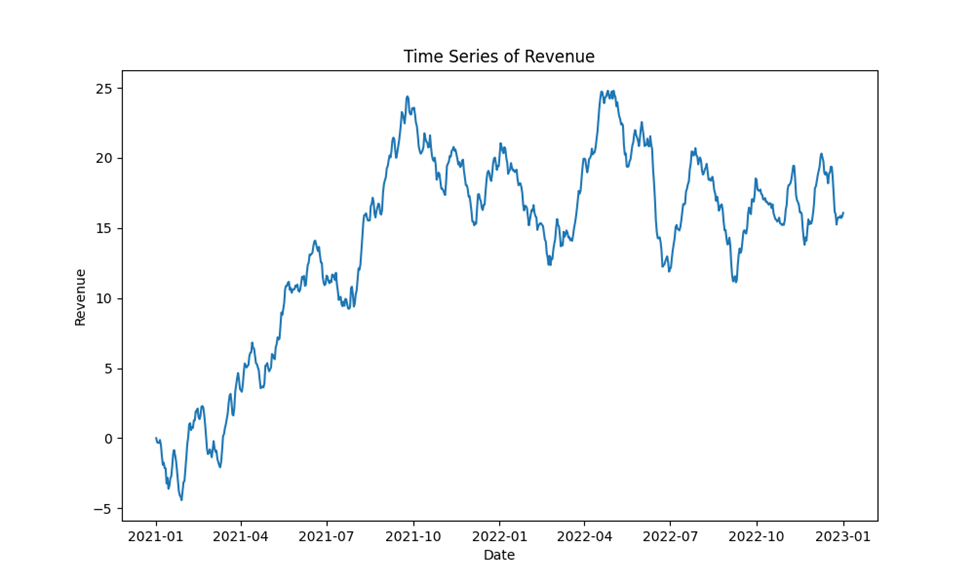


Figure 3: Line Graph of Time Series with Datetime

Subsequently, the stationarity of the time series is evaluated. This is completed by performing an Augmented Dickey-Fuller test. The results are captured in a screenshot below, which indicate that the time series is not stationary with 95% confidence.

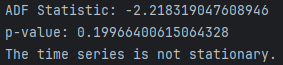


Figure 4: Screenshot of Stationarity Results

The data undergoes further cleaning and evaluation. In order to achieve stationarity, differencing was applied to the dataset. Other data manipulation techniques exist to achieve stationarity, such as log transformation, but were not suitable for this dataset due to negative values and lack of visible exponential growth. The dataset was checked for missing values and none were discovered. After the data was cleaned, differencing was applied, and the first row was dropped due to the NaN value created by differencing. The data is exported to cleaned\_processed\_data.csv. The cleaned and differenced dataset is also split 80/20 into training and testing sets, exported to train.csv and test.csv respectively. Usually the data would be split randomly, but for time series the splits contain consecutive data.

**Requirement E: Data Analysis**

After the data is preprocessed, the training set can be evaluated to determine the most appropriate ARIMA model, accounting for the observed trend and seasonality. These can be evaluated by visualizing the training data for trends, autocorrelation, spectral density, and the residuals and components of the decomposed time series.

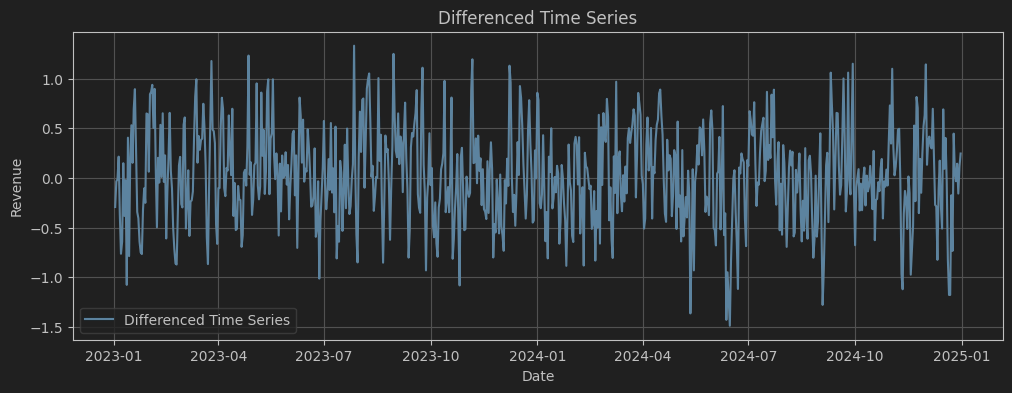


Figure 5: Plot of Differenced Data

The trend of the differenced time series is plotted in Figure 5. It can be observed that the differenced data lacks cycles or visible changes in mean and variance. This suggests that the original data is the product of a random walk. Figure 6 further illustrates this concept, showing a normal distribution among the differenced data.

A screenshot of a computer screen

Description automatically generated

Figure 6: Distribution of Values of Differenced Data

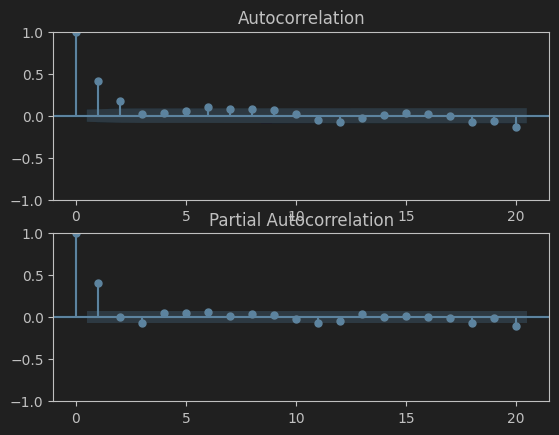


Figure 7: ACF and PCF

Figure 7 displays the autocorrelation and partial autocorrelation at different time lags. Possible values range from 1 to -1. The ACF plot can be used to estimate the optimal value for p in the ARIMA model order. The above plot suggest that AR(2) is optimal.

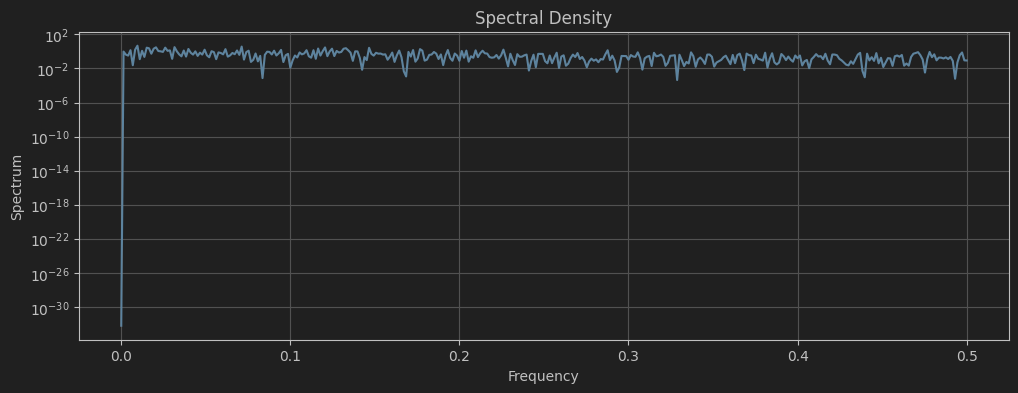


Figure 8: Spectral Density of Revenue

To address time series that display periodic behavior, a technique called spectral density can be applied to uncover underlying periodicities. Data is transformed from time domain to frequency domain, with the covariance of the time series represented by the spectral density, estimated using a periodogram. A kernel smoother can improve the interpretability of the subsequent power versus frequency plot, seen in Figure 8. The lack of spikes indicate that there is no seasonal or cyclical behavior.

Another method for analyzing time series data is through decomposition, which separates the data into a series of components: trend, seasonality, and noise. All components are visualized in Figure 8. Trend and seasonality have previously been discussed and the lack of sinusoidal waves in the third plot reinforce the previous analysis. The plot of residuals appears to be visually random, with no obvious trends or cycles and moderate, stable variability throughout. This is more obvious in Figure 10.

****

Figure 9: Plots of Components of Decomposed Time Series

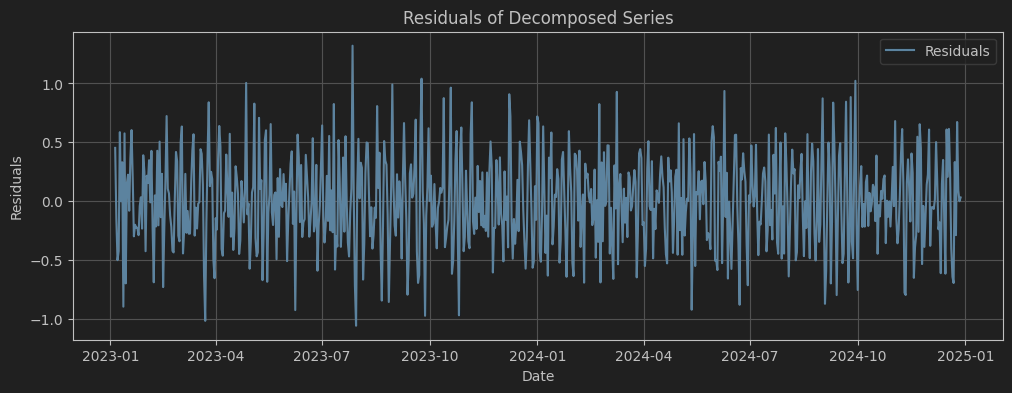
****

Figure 10: Plot of Residuals of Decomposed Series

As a result of the above analysis, it can be determined that the ARIMA model will account for the trend of the time series, without accounting for seasonality as none seems to appear in Figure 8, as the seasonal graph lacks sinusoidal curves. Using Python, I was able to automate this process using a grid search, selecting the best p, q, and d parameters based on the ARIMA model that produced the lowest Akaike Information Criterion (AIC) and later the mean absolute error (MAE). A graph showing the forecasted data is shown in Figure 11. The Python script calculations and output can be viewed in Figure 12. The grid search was performed on both the training and test data, with both forecasts subsequently scrutinized.

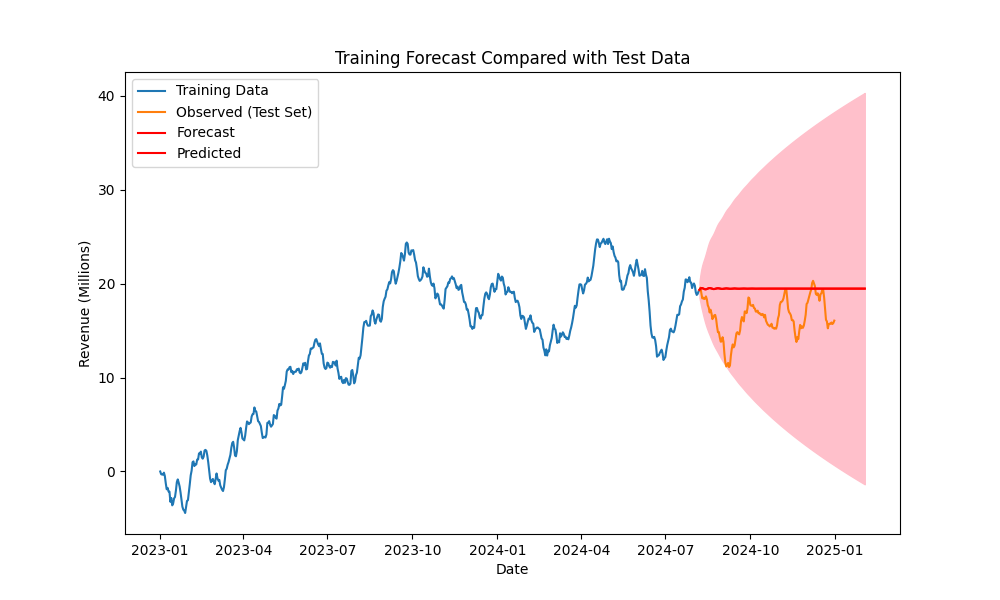


Figure 11: Revenue with Forecast

The grid search operated by defining a range of integers for each of the parameters (p, d, q). The grid search will weigh the results and select the pairing of parameters with the lowest ACI. Initially, the range of each parameter was set as 0 to 5, aside from d, which was set to 1 due to previous differencing, which resulted in training optimal parameters defined as 4, 1, and 3 and testing optimal parameters defined as 2, 1, 1. The respective ACIs for training and test are: 700.77 and 176.47. The respective MAEs for training forecast on training, test forecast on test, and training forecast on test are: 0.3525, 0.47522, and 3.072.

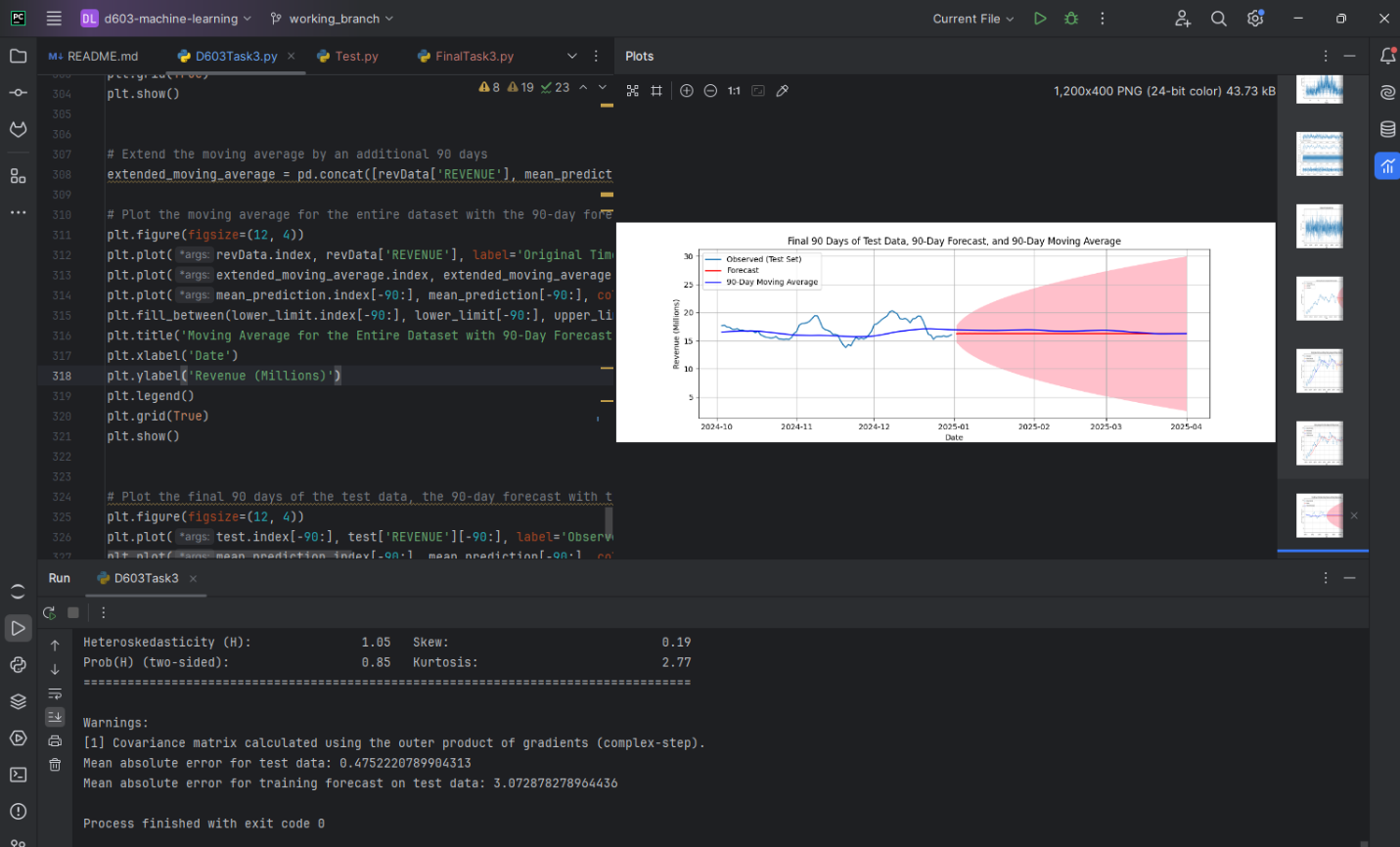


Figure 12: Output and Calculations of Optimal ARIMA

A graph with a line and a red line

Description automatically generated

Figure 13: Revenue with 90-Day Forecast

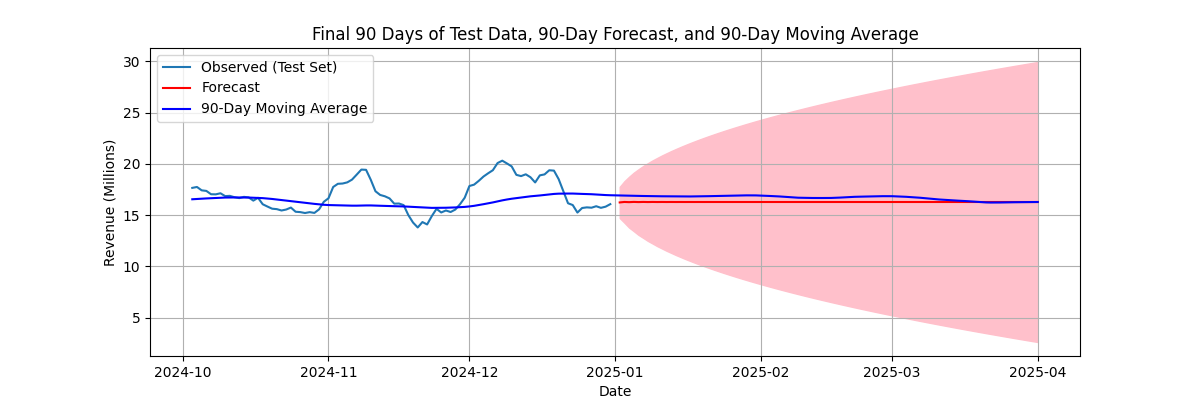


Figure 14: Closer View of 90-Days of Training Data and Forecast

**Requirement F: Summary**

As discussed previously, ARIMA modeling was chosen due to the lack of presence of seasonality within the time series. Using a grid search to find the parameters that calculated the lowest ACI value allowed for the parameter selection process to be automated. The AIC is a comparative value that penalizes for overfitting and rewards balanced, well-fitting models. The model with the lowest AIC value will have superior goodness-of-fit (Date, 2021). Seen in Figure 12, this led to parameter values of 2, 1, and 1 to be selected for p, d, and q. AutoARIMA and SARIMAX were also explored, but the best model was captured with an ARIMA grid search by comparing MAE of the forecasts and observations from ACF.

The prediction interval selected for the forecast was 90 days due to the length of the data and typical financial planning cycles. This duration allows for a meaningful analysis of trends and patterns that can impact financial decisions. By extending the forecast to 90 days, it is possible to provide a more comprehensive outlook for future revenue, one financial quarter at a time, as seen in Figure 13. This interval is particularly useful for businesses that need to plan for quarterly financial performance, budget allocations, and resource management.

The current forecast indicates that the positive effects of the readmission procedures have plateaued and are predicted to continue to plateau. This suggests that while any immediate benefits of the procedures have stabilized, there is minimal potential for continued growth in the near future, unless additional measures are taken. Short term growth is likely random. By anticipating this future plateau and evaluating the most impactful changes that initially inspired the initial positive trend, healthcare providers can strategically plan and allocate resources to further capitalize on any previous growth, ensuring sustained financial health and continued improvement in patient care.

**Requirement G: Interactive Development Environment**

<file:///C:/Users/gabri/PycharmProjects/d603-machine-learning/exportToHTML/D603Task3.ipynb.html>

**Sources Cited**

**Code:**

1. Artley, B. (2022, May 12). *Time Series Forecasting with ARIMA , SARIMA and*

*SARIMAX*. Medium. <https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6>

1. Date, S. (2021, June 2). *How To Isolate Trend, Seasonality And Noise From A Time*

*Series*. Time Series Analysis, Regression and Forecasting. https://timeseriesreasoning.com/contents/time-series-decomposition/

1. Meraxes. (2018, May 10). *Splitting time series data into train/test/validation sets*. Cross

Validated. <https://stats.stackexchange.com/questions/346907/splitting-time-series-> data-into-train-test-validation-sets

1. Pulagam, S. (2020, June 27). *Time Series forecasting using Auto ARIMA in python*.

Medium. <https://towardsdatascience.com/time-series-forecasting-using-auto-arima-in-python-bb83e49210cd>

**Report:**

1. Date, S. (2021, June 15). *The Akaike Information Criterion*. Time Series Analysis,

Regression and Forecasting. https://timeseriesreasoning.com/contents/akaike-information-criterion/

1. GeeksforGeeks. (2024, April). *Time Series Analysis and Forecasting*. GeeksforGeeks.

[https://www.geeksforgeeks.org/time-series-analysis-and-forecasting/#](https://www.geeksforgeeks.org/time-series-analysis-and-forecasting/)

1. Kumar. (2021, December 16). *How to Check if Time Series Data is Stationary with*

*Python?* GeeksforGeeks. https://www.geeksforgeeks.org/how-to-check-if-time-

series-data-is-stationary-with-python/#

1. Muralidhar, K. S. V. (2021, August 5). *Understanding Autocorrelation in Time Series*

*Analysis*. Medium. <https://towardsdatascience.com/understanding-autocorrelation-in-time-series-analysis-322ad52f2199>

1. Wang, P. (2020, November 24). *AutoCorrelation*. GeeksforGeeks.

https://www.geeksforgeeks.org/autocorrelation/