**Task 3: Time Series Modeling**

Joseph Emmanuel Cayetano

Western Governors University

D603: Machine Learning Task 3

February 16, 2025

**A. Create your subgroup and project in GitLab**

**GitLab Repository URL:** <https://gitlab.com/wgu-gitlab-environment/student-repos/jcayet5/d603-machine-learning/-/tree/task3branch?ref_type=heads>

**Repository Branch History Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

**B1. Summarize one research question that is relevant to a real-world organizational situation captured in the selected dataset and that you will answer using time series modeling techniques (REVISION FIXED)**

One research question that is relevant to a real-world organizational situation captured in the medical dataset and that I will answer using time series modeling technique is: Can hospital daily revenue be accurately forecasted compared to actual observed values?

**B2. Define the objectives or goals of the data analysis**

The data analysis has multiple objectives. The first objective is to analyze the hospital's daily revenue from the past two years to identify trends, seasonality, and fluctuations. The second objective is to build a predictive ARIMA or SARIMA model to forecast daily hospital revenue based on historical data. The third objective is to evaluate the model's forecasted values compared to the actual observed values. The fourth objective is to evaluate the model's performance using statistical metrics to ensure reliable predictions. The fifth objective is to provide actionable insights to help the hospital with resource allocation and financial planning.

**C. Summarize the assumptions of a time series model including stationarity and autocorrelated data**

The time series model has multiple assumptions. The first assumption is stationarity, meaning that the "statistical properties of a time series (or rather the process generating it) do not change over time" (Affek, 2019, par. 2). This does not mean the series is flat, but that its fluctuations or patterns remain stable over time. Many time series models assume stationarity to simplify analysis and forecasting. If a time series is not stationary, techniques like differencing may be needed. The second assumption is autocorrelation, meaning past values influence future values, making observations dependent. A time series often shows autocorrelation, where patterns repeat over time, making it more predictable. The autocorrelation assumption is important because it helps models identify patterns, such as trends and seasonality, and ensures more accurate forecasting. The third assumption is homoscedasticity, meaning the variance of errors should remain stable over time. If the variance is not stable, forecasting accuracy may decrease, requiring transformations like logarithmic scaling. The fourth assumption is uncorrelated errors, meaning that residuals (errors) should be independent and not show any patterns or correlations over time. If the errors are correlated, it could lead to biased estimates.

**D1. Provide a line graph visualizing the realization of the time series**

A graph with red lines

Description automatically generated

**D2. Describe the time step formatting of the realization, including any gaps in measurement and the length of the sequence**

Regarding the time step formatting of the realization, the x-axis represents "Day", indicating that the time series is measured at daily intervals. Each data point represents a day in the hospital's first two years of operation. The time series also appears continuous, with no visible gaps, indicating that the data was collected consistently each day. As for the length of the sequence, the time series spans 731 days, which covers two years of hospital revenue data. Regarding the gaps in measurement, the graph shows no missing days or interruptions. Each day has a revenue value, and the line graph progresses smoothly without breaks.

**D3. Evaluate the stationarity of the time series (REVISION FIXED)**

A screenshot of a computer program

AI-generated content may be incorrect.

In this code, I used the ADF test on the Revenue column to identify whether the time series is stationary. I used the adfuller() function to run the ADF test on the Revenue column and extracted its ADF statistic, p-value, and critical values.

The ADF statistic (-2.21) is higher than all the critical values at the 1%, 5%, and 10% levels. The p-value (0.19) is higher than the standard threshold of 0.05. This means the ADF test fails to reject the null hypothesis and confirms that the time series is non-stationary and requires differencing.

**D4. Explain the steps used to prepare the data for analysis, including the training and test set split (REVISION FIXED)**

A screenshot of a computer

Description automatically generated

I imported and loaded the medical dataset using panda's read\_csv() function. The output from the head() function verifies that the dataset loaded correctly. The dataset contains 731 rows (days) and two columns, which are Day and Revenue. The variable Day is a day during the hospital's first two years of operation, and the variable Revenue is the hospital's revenue in million dollars.

A computer screen shot of a computer code

Description automatically generated

I checked for missing values by using the isnull().sum() function from pandas. Since there are no missing values, we can proceed to the next step without further data cleaning.

A close up of a text

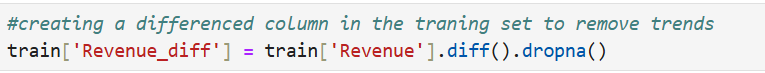
Description automatically generated

I set the variable Day as the index for time series analysis by using the set\_index() function from pandas.

A screenshot of a computer code

Description automatically generated

In this code, I split the medical dataset into training and test sets. The training set gets 80% of the data, while test set gets 20% of the data. This was done using the iloc method to preserve the sequential order of the time series. I used the shape method to confirm the split and display the training and test dataset sizes.



In this code, I created a new column called Revenue\_diff in the training dataset by applying differencing to the Revenue column. This is to remove trends and make the data stationary.

A screenshot of a computer program

AI-generated content may be incorrect.

In this code, I used the ADF test on the differenced Revenue column in the training set to verify if the time series is now stationary. I used the adfuller() function to run the ADF test on the differenced column and extracted its ADF statistic, p-value, and critical values.

The ADF statistic (-7.90) is much smaller than all critical values at the 1%, 5%, and 10% levels, which means the null hypothesis that the differenced series is non-stationary can be rejected. The p-value (4.09e-12) is extremely small, well below the standard threshold of 0.05. This provides strong evidence to reject the null hypothesis and confirms that the differenced time series is now stationary.

**D5. Provide a copy of the cleaned dataset**

A close-up of a white card

AI-generated content may be incorrect.

In this code, I used the to\_csv() function from pandas to export both cleaned training and test datasets as CSV files. I included these files in my submission.

**E1. Report the annotated findings with visualizations of your data analysis**

Trends

A graph with red lines

Description automatically generated

In this code, I created a line graph that shows the trend of the hospital revenue over time using the training set. Inside the plot() function, I used the Day index for the x-axis and the training Revenue column for the y-axis. I also added axis labels and a title.

In the graph, the hospital's revenue increases steadily in the first 300 days. Then after day 300, revenue stabilizes but starts fluctuating in cycles. Peaks and troughs repeat regularly, indicating possible seasonal patterns. Additionally, the size of these fluctuations changes over time, indicating that the variance is not constant. These factors suggest that the time series is non-stationary.

A graph of red lines

AI-generated content may be incorrect.

In this code, I created a line graph that shows the trend of the differenced Revenue data in the training set. Inside the plot() function, I used the Day index for the x-axis and the differenced Revenue data for the y-axis.

In the graph, revenue values fluctuate around zero, indicating that differencing made the data stationary. There is also no clear seasonal pattern in the graph, as differencing removes trends and seasonality.

The Autocorrelation Function

A screen shot of a graph

Description automatically generated

In this code, I created an Autocorrelation Function (ACF) plot for the Revenue data in the training set by using the plot\_acf() function. I also displayed autocorrelations up to lag 50.

In the plot, the autocorrelation starts at 1.0 and decreases slowly across lags. The slow decrease indicates that the time series is non-stationary, which means it has trends or seasonality. To make the series stationary, differencing needs to be applied. In the next plot, I performed an ACF test on the differenced Revenue column to confirm its stationarity.

A screen shot of a graph

Description automatically generated

In this code, I created an Autocorrelation Function (ACF) plot for the differenced Revenue data in the training set by using the plot\_acf() function. I also displayed autocorrelations up to lag 50.

In the plot, Lags 1 and 2 show significant positive correlation as they are well above the confidence interval. After lag 2, most bars fall within the confidence interval, indicating that differencing has removed the trend, and the time series is now stationary.

The Spectral Density

A graph with red lines

Description automatically generated

In this code, I created a spectral density plot for the Revenue data in the training set to identify dominant frequencies in the time series. I used the periodogram() function from the scipy library to compute the power spectral density (PSD) of Revenue. I also used the semilogy() function to make the small peaks more visible. Then, I plotted the frequencies on the x-axis and PSD on the y-axis.

In the plot, the spectral density is highest near frequency = 0, which means most of the variation in Revenue comes from low-frequency components. This suggests that the data is mainly influenced by long-term trends or seasonality. The spectral density slowly decreases, which means higher-frequency components have less impact on the time series.

A graph with red lines

Description automatically generated

In this code, I created a spectral density plot for the differenced Revenue data in the training set to identify dominant frequencies in the time series. I used the periodogram() function from the scipy library to compute the power spectral density (PSD) of the Revenue\_diff column. I also used the semilogy() function to make the small peaks more visible. Then, I plotted the frequencies on the x-axis and PSD on the y-axis.

In the plot, the spectral density is fairly flat across all frequencies after differencing, which means there are no strong periodic patterns remain. This suggests that differencing successfully removed trends and seasonality from the original Revenue data.

The Decomposed Time Series

A screenshot of a computer program

Description automatically generated

A graph of different types of graphs

Description automatically generated with medium confidence

In this code, I performed time series decomposition on the Revenue column of the training dataset and visualized its components: the original series, trend, seasonality, and residuals. I used the seasonal\_decompose() function from the statsmodels library to split the original time series into trend, seasonality, and residuals. I also specified a monthly seasonality by providing this parameter: period = 30. Then, I plotted each component.

In the original time series visualization, the hospital revenue fluctuates over time with an upward trend and cycles. In the trend visualization, it captures long-term revenue changes, showing steady growth with fluctuations. In the seasonality visualization, the seasonal component shows repeating patterns, confirming monthly seasonality in hospital revenue. In the residual component visualization, the residuals fluctuate around zero, which indicates that trend and seasonality have been removed. With no clear trends or repeating patterns, the residuals appear to be mostly random noise.

Confirmation of Lack of Trends in the Residuals of the Decomposed Series

A graph with lines and text

Description automatically generated

In this code, I created a plot to visualize the residuals from the decomposed time series. I plotted the residuals on the y-axis against the Day index on the x-axis. I also added a horizontal dash red line at y = 0 to help see whether the residuals are centered around zero.

In the plot, the residuals fluctuate around zero and show no clear upward or downward pattern, which means the trend was successfully removed. To confirm the lack of trend in the residuals of the decomposed series, I will perform the Augmented Dickey-Fuller (ADF) test in the next step.

A screenshot of a computer program

AI-generated content may be incorrect.

In this code, I performed the Augmented Dickey-Fuller (ADF) test on the residuals from the decomposed time series to check if the trend has been removed. I used the adfuller() function to run the ADF test on the residuals and extracted the ADF statistic, p-value, and critical values.

The ADF statistic (-7.49) is much smaller than all critical values at the 1%, 5%, and 10% levels, which means the null hypothesis that the residuals are non-stationary can be rejected. The p-value (4.42e-11) is extremely small, well below the standard threshold of 0.05. This provides strong evidence to reject the null hypothesis and confirms that the residuals are stationary. Based on these results, the residuals from the decomposed time series are stationary, which confirms that the decomposition process successfully removed the trend from the original series.

**E2. Identify an autoregressive integrated moving average (ARIMA) model that accounts for the observed trend and seasonality of the time series data (REVISION FIXED)**

Justification for Choosing Seasonal Autoregressive Integrated Moving Average (SARIMA) Modeling on the Time Series Data

I chose SARIMA for the time series data because the trend and seasonality visualizations show a clear 30-day cycle. A standard ARIMA model doesn't handle seasonality, but SARIMA does, which makes it the best choice for the time series data. The SARIMA model parameters include both non-seasonal (p, d, q) and seasonal (P, D, Q, s) components. I will determine the best values for these parameters using visualizations in the next steps.

Non-seasonal Components (p, d, q)

Identifying Autoregressive Order (p)

A screen shot of a graph

Description automatically generated

In this code, I plotted the PACF for the differenced time series to identify the best AR (p) value for SARIMA. I used the plot\_pacf() function with the Revenue\_diff column. I also set lags to 50 to analyze the first 50 lags and applied the Yule-Walker method for PACF estimation.

In the PACF plot, there is a strong spike at lag 1, which is well above the shaded confidence interval. After lag 1, most lags fall within the shaded confidence interval, which makes them statistically insignificant. Since only lag 1 is clearly significant, the best AR (p) value is 1.

Identifying Moving Average Order (q)

A screen shot of a graph

Description automatically generated

In this code, I plotted the ACF for the differenced time series to identify the best moving average (q) value for SARIMA. I used the plot\_acf() function with the Revenue\_diff column. I also set lags to 50 to analyze the first 50 lags.

In the ACF plot, there are two significant spikes at lags 1 and 2, and they are both above the shaded confidence interval. After lag 2, most lags fall within the shaded confidence interval, which makes them statistically insignificant. Since both lag 1 and lag 2 are significant, the best moving average (q) value is 2.

Identifying Differencing Order (d)

A screenshot of a computer program

AI-generated content may be incorrect.

In this code, I used the ADF test on the differenced time series to identify the best differencing order (d) value for SARIMA. I used the adfuller() function to run the ADF test on the Revenue\_diff column and extracted the ADF statistic, p-value, and critical values.

The ADF statistic (-7.90) is much smaller than all critical values at the 1%, 5%, and 10% levels, which means the null hypothesis that the differenced series is non-stationary can be rejected. The p-value (4.09e-12) is extremely small, well below the standard threshold of 0.05. This provides strong evidence to reject the null hypothesis and confirms that the differenced series is stationary. Since the Revenue\_diff column was differenced once to achieve stationarity, and the original Revenue column will be used in the SARIMAX() function, the differencing order should be set to d=1 so the model applies one difference internally. Therefore, the best differencing order (d) value is 1.

Justification for Using Seasonal Differencing (train['Revenue\_seasonal\_diff']) for Identifying Seasonal Components (P,D,Q)

I created the seasonal differenced Revenue (train['Revenue\_seasonal\_diff']) by using this code: train['Revenue\_seasonal\_diff'] = train['Revenue'].diff(30). I used seasonal differencing to identify the seasonal components (P,D,Q) of the SARIMA model because it removes seasonal trends with a period of 30 days. This ensures that the time series becomes stationary with respect to seasonality, which is required for SARIMA modeling.

Seasonal Components (P, D, Q, s)

Identifying Seasonal Autoregressive Order (P)

A screenshot of a computer screen

Description automatically generated

In this code, I plotted the PACF of the seasonally differenced series to identify the seasonal AR order (P) for SARIMA with seasonality of 30 days. I started by applying seasonal differencing by subtracting each value from its value 30 days earlier to remove seasonal trends. Then I plotted the PACF with lags set to 12 to analyze seasonal patterns over a year.

In the PACF plot, there are two significant spikes at lags 1 and 2. Lag 1 has a strong positive spike while lag 2 has a strong negative spike. Both are outside the shaded confidence interval, which makes them statistically significant. After lag 2, most lags fall within the shaded confidence interval, which makes them not statistically significant. Therefore, the best seasonal AR (P) value is 2.

Identifying Seasonal Moving Average Order (Q)

A screen shot of a graph

Description automatically generated

In this code, I plotted the ACF of the seasonally differenced series to identify the seasonal moving average order (Q) for SARIMA. I used the Revenue\_seasonal\_diff column, which removes seasonal patterns by subtracting values 30 days apart. Then I plotted the ACF with lags set to 12 to analyze seasonal patterns over a year.

In the ACF plot, all lags show strong correlations, with autocorrelation slowly decreasing over time. Lags 1 and 2 have significant spikes above the confidence interval, while after lag 2, the values start to decline. This suggests that higher MA terms add little value. Therefore, the best seasonal MA (Q) value is 2 since it captures enough seasonality while avoiding unnecessary complexity.

Identifying Seasonal Differencing Order (D)

A screenshot of a computer program

AI-generated content may be incorrect.

In this code, I used the ADF test on the seasonally differenced series to identify the best seasonal differencing order (D) for SARIMA. I used the adfuller() function to run the ADF test on the Revenue\_seasonal\_diff column and extracted the ADF statistic, p-value, and critical values.

The ADF statistic (-4.47) is much smaller than all critical values at the 1%, 5%, and 10% levels, which means the null hypothesis that the seasonally differenced series is non-stationary can be rejected. The p-value (0.00021) is extremely small, well below the standard threshold of 0.05. This provides strong evidence to reject the null hypothesis and confirms that the seasonally differenced series is stationary. Since the Revenue\_seasonal\_diff column was differenced once to achieve stationarity, and the original Revenue column will be used in the SARIMAX() function, the seasonal differencing order should be set to D=1 so the model applies one difference internally. Therefore, the best seasonal differencing order (D) value is 1.

Identifying Seasonal Period (s)

A graph with red lines

Description automatically generated

In this code, I created a plot to visualize the seasonal component of the decomposed training series to identify the seasonal period for SARIMA. I plotted the seasonal component on the y-axis against the Day index on the x-axis.

In the visualization, the seasonal pattern repeats consistently over time. The peaks and troughs occur at regular intervals, indicating a periodic pattern. Visually, the cycles repeat every 30 days, which aligns with monthly seasonality in daily data. Therefore, the best seasonal period (s) value is 30.

Seasonal ARIMA Model (SARIMA) To Use

Based on the visualizations, the best parameter values for SARIMA ARE: p = 1, d = 1, q = 2, P = 2, D = 1, Q = 2, and s = 30. Therefore, the SARIMA model that I will use for the time series analysis is **SARIMA(1,1,2)(2,1,2,30).**

**E3. Perform a forecast using the derived ARIMA model identified in part E2**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

In this code, I fit the chosen SARIMA model (SARIMA(1,1,2)(2,1,2,30)) on the entire dataset and generated a forecast for the next 100 days. I started by defining the SARIMA model's non-seasonal and seasonal component parameters. Then, I used the SARIMAX() function from statsmodels library to train the model on the entire dataset. Next, I used the get\_forecast() function on the trained SARIMA model to generate a revenue forecast for the next 100 days. I also extracted the model's predicted mean values and confidence intervals for visualization. Then, I plotted the actual and forecasted values.

In the SARIMAX model summary, 'ar.L1', 'ma.L1', and 'ma.L2' represent non-seasonal AR and MA components, while 'ar.S.L30', 'ar.S.L60', 'ma.S.L30', and 'ma.S.L60' are their seasonal counterparts. Only the seasonal AR and MA components (ar.S.L30, ma.S.L30, and ma.S.L60) have p-values below 0.05, indicating they are statistically significant. The model has a log likelihood value of -416.16, AIC value of 848.31, and BIC value of 883.97. The Ljung-Box (Prob(Q)) and Jarque-Bera (Prob(JB)) tests return p-values of 0.91 and 0.62, both above 0.05, indicating that the residuals are normally distributed and show no significant autocorrelation. Overall, the SARIMA model appears to fit the data well based on these results.

The forecast visualization includes the historical revenue data (blue line), the SARIMA model's forecast for the next 100 days (red line), and a confidence interval (shaded pink area) representing prediction uncertainty. The SARIMA model predicts that revenue will slowly increase over the next 100 days. This closely follows the previous trends, indicating that the model captures the pattern well. Also, the confidence interval widens over time, indicating higher uncertainty in long-term predictions.

**A screenshot of a computer

AI-generated content may be incorrect.**

The forecast mean results show that there is a steady revenue increase over the next 100 days, with values starting at around 16.3 million dollars and slowly rising to around 18.4 million dollars by the end of the forecast period. This suggests that the SARIMA model predicts a consistent upward trend, aligning with past revenue patterns.

**E4. Provide the output and calculations of the analysis you performed**

A screenshot of a computer

AI-generated content may be incorrect.

In the SARIMAX model summary, 'ar.L1', 'ma.L1', and 'ma.L2' represent non-seasonal AR and MA components, while 'ar.S.L30', 'ar.S.L60', 'ma.S.L30', and 'ma.S.L60' are their seasonal counterparts. Only the seasonal AR and MA components (ar.S.L30, ma.S.L30, and ma.S.L60) have p-values below 0.05, indicating they are statistically significant. The model has a log likelihood value of -416.16, AIC value of 848.31, and BIC value of 883.97. The Ljung-Box (Prob(Q)) and Jarque-Bera (Prob(JB)) tests return p-values of 0.91 and 0.62, both above 0.05, indicating that the residuals are normally distributed and show no significant autocorrelation. Overall, the SARIMA model appears to fit the data well based on these results.

A graph with blue lines and red text

AI-generated content may be incorrect.

The forecast visualization includes the historical revenue data (blue line), the SARIMA model's forecast for the next 100 days (red line), and a confidence interval (shaded pink area) representing prediction uncertainty. The SARIMA model predicts that revenue will slowly increase over the next 100 days. This closely follows the previous trends, indicating that the model captures the pattern well. Also, the confidence interval widens over time, indicating higher uncertainty in long-term predictions.

A screenshot of a computer

AI-generated content may be incorrect.

The forecast mean results show that there is a steady revenue increase over the next 100 days, with values starting at around 16.3 million dollars and slowly rising to around 18.4 million dollars by the end of the forecast period. This suggests that the SARIMA model predicts a consistent upward trend, aligning with past revenue patterns.

A screenshot of a computer screen

AI-generated content may be incorrect.

The forecast confidence interval results show that the lower and upper bounds widen over time, indicating greater uncertainty in longer forecasts. It starts narrow, showing higher confidence in short-term predictions, but expands as the forecast extends. This should be expected, as long-term predictions are less reliable.

A screenshot of a computer code

AI-generated content may be incorrect.

In this code, I trained the SARIMA(1,1,2)(2,1,2,30) model on the training Revenue data and evaluated its performance using the Root Mean Squared Error (RMSE) on the test dataset. I started by fitting the model on the training Revenue data. Next, I used the get\_forecast() function on the trained SARIMA model to generate future predictions for the test period. Then, I extracted the model's predicted values for comparison with actual test values.

I used the Root Mean Squared Error (RMSE) metric to measure how well the SARIMA model's forecasts align with actual test data. The model has an RMSE of 4.54, which means the model's forecasts deviate from the actual revenue by about 4.54 million dollars. This indicates that the SARIMA model is a reasonably good model for capturing trends in Revenue data.

**F1. Discuss the results of your data analysis**

I selected the SARIMA(1,1,2)(2,1,2,30) model based on trend and seasonality patterns in hospital revenue data. The PACF plots identified the autoregressive orders, with p=1 for the non-seasonal component and P=2 for the seasonal component. The ACF plots identified the moving average orders, with q=2 for the non-seasonal component and Q=2 for the seasonal component. The ADF test verified that one order of differencing (d=1, D=1) was sufficient to make the data stationary. The seasonal period (s=30) was determined based on seasonal decomposition and knowledge about monthly cycles. This is the SARIMA model that was trained and evaluated.

Regarding the prediction interval of the forecast, the forecasted revenue for the next 100 days shows a steady increase, with prediction intervals widening over time. At the start of the forecast, the interval is narrow, which means the model is more certain about short-term revenue trends. However, by the end of the 100-day forecast period, the interval has widened significantly because the model is more uncertain about long-term revenue trends. This should be expected, as long-term predictions are less reliable.

I chose a 100-day forecast to balance accuracy and long-term planning. A shorter period, like 30 days, might not capture overall revenue trends, while a longer period, like 180 days, would be less reliable due to increasing uncertainty, as seen in the widening prediction interval. Also, since the dataset covers two years of data, it provides enough data to support a reliable 100-day forecast. Additionally, hospitals typically operate on monthly and quarterly financial cycles, which makes a 100-day forecast practical for decision making.

The SARIMA model was evaluated using multiple approaches. The first approach involves visually inspecting forecast accuracy. The forecast visualization shows that the model effectively captures the overall revenue trend. The forecasted values align well with actual test data, though they deviate slightly during sharp fluctuations. The second approach involves analyzing the prediction interval. The visualization shows that the prediction interval widens overtime, indicating greater uncertainty in longer forecasts. Most test data points fall within the interval, indicating that the model's uncertainty estimates are reasonable, and the forecast is generally reliable. The third approach involves analyzing the residuals. The Ljung-Box (Prob(Q)) and Jarque-Bera (Prob(JB)) tests both have p-values above 0.05, indicating that the residuals are normally distributed and show no significant autocorrelation, further supporting the SARIMA model's reliability. The Root Mean Squared Error (RMSE) is the error metric used to evaluate how accurately the SARIMA model's forecasts match the actual test data. The model's RMSE of 4.54 indicates that its forecasts deviate from the actual revenue by around 4.54 million dollars on average. This suggests that the SARIMA model is reliable and reasonably accurate in predicting hospital revenue trends.

**F2. Provide an annotated visualization of the forecast of the final model compared to the test set**

A computer code with many text

AI-generated content may be incorrect.

A graph showing a line

AI-generated content may be incorrect.

In this code, I fit the chosen SARIMA model (SARIMA(1,1,2)(2,1,2,30)) on the training set and generated a forecast. I started by defining the SARIMA model's non-seasonal and seasonal component parameters. Then, I used the SARIMAX() function from the statsmodels library to train the model on the training set. Next, I used the get\_forecast() function on the trained SARIMA model to generate future predictions. I also extracted the model's predicted mean values and confidence intervals for visualization. Then, I plotted the forecast alongside the actual test data for comparison, including a confidence interval to show uncertainty.

The forecast visualization includes training data (blue line) for past revenue trends, actual test data (green line) for comparison, SARIMA forecast (red line) for predicted values, and a confidence interval (shaded pink area) representing prediction uncertainty. Overall, the SARIMA model effectively captures the general trend in revenue. The forecasted values align reasonably well with actual test data, but are slightly off during sharp fluctuations. Most test data points fall within the confidence interval, which confirms that the SARIMA model's uncertainty estimates are reasonable, and the forecast is generally reliable.

A screenshot of a computer program

AI-generated content may be incorrect.

**A graph showing a line graph

AI-generated content may be incorrect.**

The forecast visualization includes the training data (blue line), actual test data (green line), the SARIMA model's forecast for the next 100 days (red line), and a confidence interval (shaded pink area) representing prediction uncertainty. The SARIMA model predicts that revenue will slowly increase over the next 100 days. This closely follows the previous trends, indicating that the model captures the pattern well. Also, the confidence interval widens over time, indicating higher uncertainty in long-term predictions.

**F3. Recommend a course of action based on your results**

There are multiple courses of actions that the hospital can take based on the time series results. First, the hospital should use the forecasted revenue trends for budgeting and resource allocation to keep financial decisions in line with expected income. Second, since the prediction intervals widen over time, showing more uncertainty in long-term forecasts, the hospital should prioritize short-term forecasts for immediate decisions and use the full 100-day forecast for strategic planning. Third, the hospital can use the prediction interval's upper and lower bounds to prepare for best and worst case revenue scenarios and develop contingency plans for unexpected revenue declines.

**G. Include a PDF or HTML document of your executed notebook presentation**

The D603\_Task3\_Final.html is the HTML file of this jupyter notebook. I included it in my submission.

**H. Record the web sources used to acquire data or segments of third-party code to support the analysis**

Andres, D. (2023, June 15). *Step-by-Step Guide to Time Series Forecasting with SARIMA Models*. Machine Learning Pills. <https://mlpills.dev/time-series/how-to-train-a-sarima-model-step-by-step/>

**I. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized**

Affek, S. P. (2019, November 12). *Detecting stationarity in time series data*. Medium. <https://medium.com/towards-data-science/detecting-stationarity-in-time-series-data-d29e0a21e638>