WGU MSDA Program

Time Series Analysis of Hospital System Revenue

Write Up

Christopher Kamper

February 21st, 2024

**Part I: Research Question**

1. 1. Can the hospital's daily revenue for January of 2023 be effectively forecasted based on the actual values from the first two years of operation?

* 1. The goal of this analysis is to create a model to predict future revenue for the hospital system. Daily revenue numbers are provided in the data set from the first two years of operation. This will allow for a model to be trained and tested using this data to forecast future values. First, exploration and cleaning will verify and help ensure that the data is ready for a time series analysis. After all necessary checks have been completed, a test model will be run on training data before being evaluated. This model will then be used to predict the revenue forecast.

**Part II: Method Justification**

A time series analysis needs to run under the assumption that the data is considered stationary. Stationary is defined as a data set whose statistical properties do not change over time. This includes the mean, variance, and autocorrelation structure. This assumption is needed as many models, including ARIMA, function by thinking the patterns and relationships in the data are stable. (Ravelo)

ARIMA is a great option for this analysis as long as the data can be shown to not include any seasonal trends. If this is true, ARIMA can work by taking the actual values provided in the data set to evaluate and train a model that will predict future values. This is accomplished as ARIMA is an autoregression method that shows how a variable changes over time-based on its own lagged values. (“ARIMA Modeling”)

**Part III: Data Preparation**

1. 1. Line graph of the original time series:

A graph showing a line

Description automatically generated with medium confidence

1. This data set was provided with a column showing the revenue for a numeric valued day of operation. For this analysis, the data was prepared by setting day 1 as 01-01-2021. This added an actual date to each value and was then used to set the index for the data set. This allows for easy reading of the graphs to determine the timeframe of the revenue numbers used for the model as well as what was predicted by the model.

The data set shows there are 731 rows. Since the data set was set to start on 01-01-2021, the last date covered is 01-01-2023. This shows the full two years of observations that the dictionary referenced so it can be assumed there are no missing values.

If there were any, one could fill in an average value to retain the dates for the time series without too much concern over having a significant effect on the data set. With this set, an average value for each month could easily be found and inputted. In other sets, it may be best to look at other options to input values.

1. The original time series was found to not be stationary based on the results of the Dickey-Fuller test as the p-value calculated was greater than 0.05. There also was visual evidence of this seen in the Original Series with Rolling Mean and Standard Deviation graph shown below.

A graph showing a graph

Description automatically generated with medium confidence

To get the data stationary, differencing was used. This method calculates the differences between each lag in the data set to create a stationary set. This is due to the method stabilizing the mean through the removal of the changes in the level of a time series. (“8.1 Stationarity and Differencing | Forecasting: Principles and Practice (2nd Ed)”)

1. Cleaning steps used after loading data:
   1. Identify and drop duplicated values based on the "Day" column.
   2. Count the number of null values in the Data Frame and sum by column.
   3. Convert the "Day" column to reflect actual date values and set as the index for the Data Frame.
   4. Drop the "Day" column since the new index will take care of that.
   5. Since the data was found to be not stationary, use the differencing to make the data stationary.

1. Copy of the cleaned data set is attached.

**Part IV: Model Identification and Analysis**

1. 1. Visualizations
      1. There is some seasonality shown on the Decomposition Seasonal Zoom plot from the decomposition. This appears to be in the form of a high-frequency pattern that could imply it being weekly. Since this is a small time frame compared to the larger sample size and the magnitude is very small, it is not expected that this would have a strong affect. Taking this into consideration with other metrics and stationarity confirmed, an AR model will be used for this analysis.

A blue lines with black text

Description automatically generated

1. There are no obvious trends shown based on the Stationary Series with Rolling Mean and Standard Deviation plot:

A graph showing a graph of a graph

Description automatically generated with medium confidence

1. Here is the ACF plot. This is a visual showing the values from the autocorrelation function that shows relationships between values and their lags. The steep drop off shown signifies an AR (auto regression) model should be used for the analysis whereas a more minor drop would lead one to look at a MA (moving average) model. The blue area shows the confidence interval noted in the code to let one know if a value is statistically significant or not.

A graph with blue dots and numbers

Description automatically generated

Here is the PACF plot. This is a visual showing the values from the partial autocorrelation function that shows relationships between values and a specified lag. Just like the ACF graph, the steep drop off recommends using an AR model. The blue area shows the confidence interval noted in the code to let one know if a value is statistically significant or not.

A graph with blue dots and numbers

Description automatically generated

1. Here is the spectral density plot. It shows the frequency of the values from the data set. This can help determine seasonality and model selection. This is done by looking at the patterns of how frequent a value is. Higher frequency peaks can be related to strong autoregression so it signifies an AR model should be used.

A graph showing a number of blue lines

Description automatically generated

1. Here is the decomposed time series:

A graph showing a graph of a graph

Description automatically generated with medium confidence

1. There are no obvious trends shown based on the Decomposition Resid Zoom plot from the decomposition:

A blue lines on a white background

Description automatically generated

1. The code used to create the ARIMA model is below:

# Define the p, d, and q values for the model from previous tests.

p = 1

d = 1

q = 0

# Set up the ARIMA model.

test\_model = ARIMA(train\_data["Revenue"], order=(p, d, q))

# Fit the model.

test\_model = test\_model.fit()

# Review the model summary.

test\_model.summary()

1. The code used to forecast based on the ARIMA model is below:

# Set the prediction model.

predict\_model = ARIMA(med\_time\_series['Revenue'], order=(1,1,0))

# Fit the prediction model to the data.

predict\_model = predict\_model.fit()

# Define future dates and set up an index.

index\_future\_dates = pd.date\_range(start='2023-01-01', end='2023-01-31')

print(index\_future\_dates)

# Make future predictions.

predict = predict\_model.predict(start=len(med\_time\_series), end=len(med\_time\_series)+30, typ='levels').rename('Future ARIMA Forecasted Revenue')

# Set index.

predict.index=index\_future\_dates.rename('Date')

# Define columns.

predict\_data = pd.DataFrame(forecast)

predict\_data.columns = ['Revenue']

print(predict\_data)

1. Output and calculations from analysis:

DatetimeIndex(['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',  
 '2023-01-05', '2023-01-06', '2023-01-07', '2023-01-08',  
 '2023-01-09', '2023-01-10', '2023-01-11', '2023-01-12',  
 '2023-01-13', '2023-01-14', '2023-01-15', '2023-01-16',  
 '2023-01-17', '2023-01-18', '2023-01-19', '2023-01-20',  
 '2023-01-21', '2023-01-22', '2023-01-23', '2023-01-24',  
 '2023-01-25', '2023-01-26', '2023-01-27', '2023-01-28',  
 '2023-01-29', '2023-01-30', '2023-01-31'],  
 dtype='datetime64[ns]', freq='D')

Revenue  
Date   
2023-01-01 16.171559  
2023-01-02 16.213862  
2023-01-03 16.231384  
2023-01-04 16.238642  
2023-01-05 16.241649  
2023-01-06 16.242894  
2023-01-07 16.243410  
2023-01-08 16.243623  
2023-01-09 16.243712  
2023-01-10 16.243749  
2023-01-11 16.243764  
2023-01-12 16.243770  
2023-01-13 16.243773  
2023-01-14 16.243774  
2023-01-15 16.243774  
2023-01-16 16.243774  
2023-01-17 16.243774  
2023-01-18 16.243774  
2023-01-19 16.243774  
2023-01-20 16.243774  
2023-01-21 16.243774  
2023-01-22 16.243774  
2023-01-23 16.243774  
2023-01-24 16.243774  
2023-01-25 16.243774  
2023-01-26 16.243774  
2023-01-27 16.243774  
2023-01-28 16.243774  
2023-01-29 16.243774  
2023-01-30 16.243774  
2023-01-31 16.243774

A graph with green line

Description automatically generated

1. The code used to forecast based on the ARIMA model is below:

# Set the prediction model.

predict\_model = ARIMA(med\_time\_series['Revenue'], order=(1,1,0))

# Fit the prediction model to the data.

predict\_model = predict\_model.fit()

# Define future dates and set up an index.

index\_future\_dates = pd.date\_range(start='2023-01-01', end='2023-01-31')

print(index\_future\_dates)

# Make future predictions.

predict = predict\_model.predict(start=len(med\_time\_series), end=len(med\_time\_series)+30, typ='levels').rename('Future ARIMA Forecasted Revenue')

# Set index.

predict.index=index\_future\_dates.rename('Date')

# Define columns.

predict\_data = pd.DataFrame(forecast)

predict\_data.columns = ['Revenue']

print(predict\_data)

**Part V: Data Summary and Implications**

1. 1. The result of this analysis does forecast values for the hospital system's revenue for the month of January 2023 based on the previous two years of operation. However, it is held back due to the small amount of data that is available to create the model. These predictive values are affected as they do fall into a strong confidence interval but do converge to a calculated mean as time goes on. This signals that while the predictions may be "safe", there is room for more accuracy. To improve this, more data would be needed to feed into the training of the model. Part of this is due to the limited time that the system has been operational and the drastic growth in revenue during the first year as the business ramped up. This will affect the model and its ability to accurately predict since that type of growth may not be true over a long period of time.

The ARIMA model method was chosen for this analysis due to it being able to predict future values based on past values. Since the data set provided gave the past values, this is a great method to look into using. To use ARIMA, the data must not be seasonal and must be from constant, regular intervals and this set meets that criteria. (“ARIMA Modeling”)

The prediction interval used was 30 days. This was chosen as a starting point due to the limited amount of data that the model could be trained with. During the analysis, it was noticed that even with this smaller interval, the predicted values converged to a single value so anything further would match that. The Predicted Future Revenue graph shows this as the values starting at 2023-01-14 through 2023-01-31 all predict the same number.

Auto ARIMA was used to find the ideal p, d, and q values for the model. The result game (1, 1, 0) as the values due to that being the model with the lowest AIC. Manual analysis was run on 3 other manual variants of the model that were not included in the auto method to see if there was a better AIC. The results for all three showed a larger value so (1, 1, 0) was chosen due to having the lowest AIC of all model variants run.

The model was evaluated using the root mean squared error (RMSE) method which came out to 3.5918. The test mean was 16.4148 so the model isn't necessarily the most accurate and should be used as a stand-alone metric, but it isn't off so nothing should be taken from it.

1. Graph showing the training, test, and forecasted values along with predicted confidence intervals:

A graph showing a line of blue and orange lines

Description automatically generated with medium confidence

1. It is recommended that the system use these forecasted values as a good guide to evaluate how revenue is performing through the month, but it should be noted that actual values should be expected within the confidence interval shown on the graph above. As previously stated, more data would help predict these revenue numbers better and lead to more actionable recommendations. While further analysis with more data would be best, the system can take away that the predicted values do represent an average that would need to be outperformed to continue the growth of revenue.

It would also be recommended that the system look into further reporting that could present the revenue numbers broken down by regions or procedures to dive further into the performance of the business to determine where focus should be put to drive improved revenue.

**Part VI: Reporting**

1. Jupyter Notebook PDF file is attached.

1. Web Sources for Code:

GfG. “Python ARIMA Model for Time Series Forecasting.” *GeeksforGeeks*, GeeksforGeeks, 7 Feb. 2020, [www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/](http://www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/). Accessed 18 Feb. 2024.

GfG. “Plot the Power Spectral Density Using Matplotlib Python.” *GeeksforGeeks*, GeeksforGeeks, 12 Apr. 2020, [www.geeksforgeeks.org/plot-the-power-spectral-density-using-matplotlib-python/](http://www.geeksforgeeks.org/plot-the-power-spectral-density-using-matplotlib-python/). Accessed 18 Feb. 2024.

“Plotting Dynamic Forecasts | Python.” *Datacamp.com*, DataCamp, 2023, campus.datacamp.com/courses/arima-models-in-python/fitting-the-future?ex=9. Accessed 18 Feb. 2024.

1. Sources:

“8.1 Stationarity and Differencing | Forecasting: Principles and Practice (2nd Ed).” *Otexts.com*, 2024, otexts.com/fpp2/stationarity.html. Accessed 18 Feb. 2024.

“ARIMA Modeling.” *CORP-MIDS1 (MDS)*, 15 Dec. 2023, [www.mastersindatascience.org/learning/statistics-data-science/what-is-arima-modeling/](http://www.mastersindatascience.org/learning/statistics-data-science/what-is-arima-modeling/). Accessed 18 Feb. 2024.

Ravelo, Ciera. “The Stationary Data Assumption in Time Series Analysis.” *Statistics Solutions*, 31 Jan. 2023, [www.statisticssolutions.com/stationary-data-assumption-in-time-series-analysis/](http://www.statisticssolutions.com/stationary-data-assumption-in-time-series-analysis/). Accessed 18 Feb. 2024.