**Performance Assessment D213: Time Series Modeling**

**A. Research Question**

**1.** For this assessment, the research question is as follows: using time series analysis, can we predict the next month (30 days) of revenue?

**2.** The goal of this analysis is to determine if, using time series analysis, we can use an ARIMA model to forecast upcoming revenue over the course of the next month, or 30-day period.

**B. Method Justification**

**1.** Time series analysis assumes that the data within them is stationary. This means that the values of the data do not rely or depend upon the exact time at which the data was recorded. This means that any statistical property, such as the mean and any autocorrelations, are constant over time.

The next assumption of time series is one of autocorrelation. Time series assumes that the data points will not be correlated with the previous observations over a period of time (Dobbins et al, 2021). What this is stating is that the data within the time series has a significant level of autocorrelation then that is helpful in predicting the current and future values (Prabhakaran, 2023).

**C. Data Preparation**

**1.** The following image shows a line graph for the realization of the time series:

A graph showing the value of a stock market

Description automatically generated

**2.** The time series (as seen in the line graph above) depicts daily revenue over the course of 731 days, starting at 0 units from day 1. The time series spans a total of 731 days and there are no gaps in the measurement/sequence. The following screenshot showcases the code using python that was used to determine the length of the sequence and whether there were any gaps or not:

A white background with black text

Description automatically generated

This confirms the length of the sequence while also confirming no NAs or missing days. The last step in cleaning the data is to verify that there are also no NAs within the Revenue. The following screenshot confirms that there are no NAs in our medical revenue:

A close up of a message

Description automatically generated

Now that the data has been verified to contain no NAs in either the Day column or the Revenue column, it can then be written to a CSV file. The data is now cleaned and has been attached alongside this written assessment.

**3.** In order to evaluate the time series to determine if it is stationary or not, we do what is called the ADF, or Augmented Dickey-Fuller Test. The evaluate stationary using this test, you run the test over the variable that is changing over time (in this case, revenue) and observe the p-value. If the p-value is less than 0.05 then the data can be considered stationary. If the p-value is greater than 0.05, then you must apply certain methods in order to make the data stationary. The following screenshot showcases the ADF test being used in python and the resulting p-value:

A screenshot of a computer code

Description automatically generated

The resulting p-value is 0.1997 which is above the stationary threshold of 0.05. This would indicate that the data is **not** stationary.

**4.** In order to turn the data into one that is stationary, we use a technique that is referred to as differencing. “Differencing is a technique to transform a non-stationary time series into a stationary one. It involves subtracting the current value of the series from the previous one, or from a lagged value” (Enemuo et al., 2024). To do this with our data, I created a new column called Revenue\_diff (which would be the revenue after differencing) and used the “diff” command in python to difference the revenue, while adding a 0 at the beginning in order to keep the same length. After doing this, I created a subset to remove the original revenue and only keep the differenced revenue and then rewrote the data to remove any NAs that resulted from the difference command. After doing so I re-evaluated the Revenue\_diff column by repeating the ADF test. The following screenshot shows the results:

A screenshot of a computer code

Description automatically generated

By repeating the ADF test, the resulting p-value is now less than 0.01. This indicates that the data is now stationary after differencing since the p-value is less than 0.05. The last step to prepare the data is to write the differenced data to a new CSV file. The training and testing data sets will be created during part **D.**

**5.** All data sets as well as the code to prepare the data for analysis have been attached alongside this written assessment.

**D. Model Identification and Analysis**

**1.** Now that the data is stationary due to differencing, the data can now be decomposed. By decomposing the data, this will allow us to observe key features within the data set, such as seasonality. Before decomposing the data, it was first converted into a time series object, using an arbitrary date for the day field (in this case 2020-01-01). This date will be used as the starting date for the rest of the analysis. The data was then decomposed in python and the plot of the decomposition is shown below:

A graph of blue lines

Description automatically generated with medium confidence

The decomposed time series shows us the trend component, the seasonal component, as well as the residuals. Let’s observe the information more closely, one at a time starting with the trends.

The image below showcases the trend of the decomposed time series:

A graph showing a trend

Description automatically generated

As you can see the trend component is dramatically fluctuating starting from the highest amount of trend at the very beginning and making its way below the 0 line over time. This decreasing element shows that there are no trends in the data, however, it is important to note that differencing itself is designed to remove trends from the data so it is actually expected that there would be no trends. Another way we can confirm the lack of trends is by performing the ADF test again, but on the residuals of the decomposed time series. This is shown below:

A screenshot of a computer code

Description automatically generated

The resulting p-value from running the ADF test on the residuals in less than 0.01 which itself is less than 0.05. Since the p-value is statistically significant, this also confirms the lack of trends in our decomposed time series object.

Next let’s look at the seasonal component of the decomposed time series, as seen in the plot below:

A graph showing a graph of a diagram

Description automatically generated with medium confidence

The seasonal component shows heavenly fluctuations over the course of the entire time series. This falls in line with there not being in seasonality. For there to be seasonality, it should show fluctuations over distinct periods of time, such as months, quarters, years, etc., but this seasonality component does not suggest that there is any seasonality. Let’s explore this further. Prior to running my predictions using an ARIMA model, I needed to determine the appropriate fit. I used a stepwise approach using the auto\_arima function on the ORIGINAL (not differenced) data, which is shown in the following screenshot:

A screenshot of a computer

Description automatically generated

As you can see, the stepwise approach shows that the best fit is an ARIMA model of order 1,1,0. The numbers mean the best model involves 1 autoregressive term, a differencing of lag 1 (which we already did previously for the decomposition), and a moving average of 0. The important thing to observe here is the second part of the best model indicator which says (0,0,0). This part of the best model indicator confirms what we saw earlier, which is that there is no seasonality within our data.

Next, the following screenshot shows the autocorrelation function:

A graph showing a function

Description automatically generated

The autocorrelation function here shows a peak at the first lag and then the rest seemingly rotates around 0. This suggests a strong correlation at the first lag, or observation, and then no correlation for the remaining lags and observations. This rapid decay indicates that the memory of the series is short, primarily depending only on the immediate past value, which makes sense when talking about revenue. As the revenue either grows or shrinks based on the previous day, and has no relation to the revenue from days, weeks, months, etc. prior.

Lastly, the following screenshot shows the spectral density of the differenced data:

A graph showing a number of blue lines

Description automatically generated

A spectral density plot is used to determine how powerful the variance of the data is at different frequencies. This spectral density plot shows the highest peak at a very low frequency, closer to 0. There are several other peaks through the frequencies as well and all of these together suggest that there isn’t any seasonality, as we stated before. For seasonal data, there would be peaks at distinct frequencies that suggest a clear pattern, but not in seen here, which is another confirmation of no seasonality.

**2-4.** As stated in the previous step, a stepwise approach was used to determine that the best fitted model is an ARIMA model of order 1,1,0, but before we can make predictions using that order, we first need to create our training and testing sets. The goal of our analysis is to predict the medical revenue over the next monthly period (30 days), so the training and testing data were created to include just the last 30 days of the data within our testing data. The following code shows creating the training and testing data sets for our predictions:

A screenshot of a computer program

Description automatically generated

Now that the training and testing data sets have been created (this satisfies part **C4**), the next step is to create our ARIMA model using the revenue within the training set and the order of 1,1,0. The model summary is shown below:

A screenshot of a computer

Description automatically generated

Now that the initial model has been created, the next step is to use the test data set in order to make predictions. Since we will later be forecasting the next 30 days of revenue, our predictions will be predicting the last 30 days. By doing so in python we get a predicted plot that looks like this:

A graph showing the price of the stock market

Description automatically generated with medium confidence

In this plot, the yellow line is the actual values for the revenue in the testing data set and the blue line is the mean of the predicted values. The actual values show strong revenue and the beginning of the month and then a steep drop off near the end of the month. The predictions do not seem to capture those fluctuations at all, which seem to contribute to an even prediction of revenue. The following image shows some measures of our predictions:

A screenshot of a computer

Description automatically generated

The first value is showing the actual mean of the revenue in the testing data. This states that in our testing data, the revenue averages approximately 17.9236 units. The second metric is the root mean squared error from our predictions, which comes out to approximately 1.7677 units. What this means is that the predicted values are, on average, 1.7677 units away from the actual value. This is relatively good, especially when the average is 17.9236 units, which is quite high. It is possible the dramatic drop off in the middle of the data “canceled out” the high revenue at the beginning of the data, which could explain why the predictions are mostly a constant value, but it is worth noting that the predictions were not able to capture those fluctuations precisely. But overall, a RMSE of 1.7677 is quite good, which indicates that the ARIMA model was a good fit.

Now that the initial model has been completed and evaluated, the next step is to create a second model that will be used to forecast the next 30 days of revenue. Since the start date was arbitrarily set to 2020-01-01 (or January 1st, 2020), the predictions will cover the month of January during the year 2022. To begin, I created a second ARIMA model using the full data with the same 1, 1, 0 order then created an index of the future date set, and ultimately, ran a prediction algorithm to predict the next month of revenue. When plotted, the forecasted revenue looks as follows:

A graph showing the difference between the prediction and the prediction

Description automatically generated with medium confidence

The forecast suggests a sharp increase in revenue at the beginning of the month followed by a period of constant, or sustained, revenue throughout the rest of the month, at approximately 16.24 units of revenue.

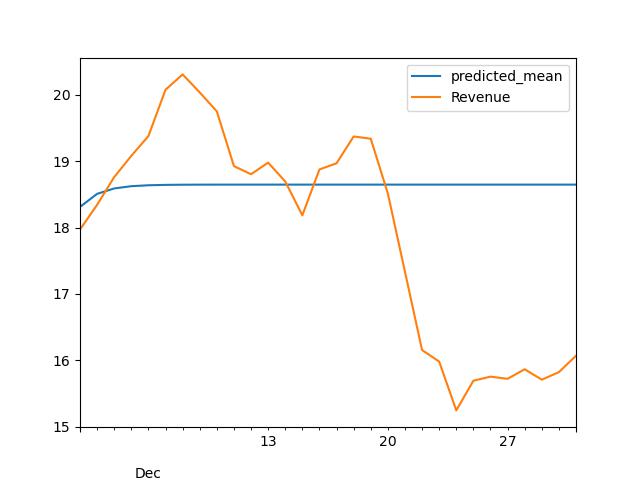
**5.** The code used for the analysis has been attached as a .py file alongside this written assessment.

**E. Data Summary and Implications**

**1.** In summary:

* A stepwise approach was used to determine the appropriate selection of an ARIMA model to predict medical revenue. This approach determined that the appropriate model was one of order 1, 1, 0. This means that the model of best fit is one with 1 autoregressive term, differenced data of lag 1, and 0 moving average.
* The forecast was created to forecast the next month of medical revenue. Since we are predicting the next month of revenue, that is the justification for a forecasted length of 30, or 30 days, which is what the testing data was used to predict. This was also the basis of the prediction interval as the testing data was created to include just the last 30 days of data in our initial data set. This is how the model was trained in order to predict revenue during the last month, which was then used to create forecast of the next month as a 30 day forecast length.
* The model was evaluated by calculating the root mean squared error to determine to accuracy of the models predictions using the testing data set. This calculation came out to an RMSE of approximately 1.7677. This indicates that the testing data predictions were within 1.77 units of the actual revenue. This is relatively close to the true value, which in our terms, indicates that the model is a decent fit, although it does not appear to capture extreme fluctuations very well. This could have affected the accuracy, but more tests need to be run to determine if that is the case or not.

**2.** As shown prior in part **D**, the following is an annotated visualization of the forecast of the predictions from the test data set compared to the actual data:



To summarize: the model predicted a near constant revenue with an approximate mean of 17.92 and a RMSE of 1.7677. It appears that the model isn’t able to account for the sharp drop in revenue that occurs within the middle of the month as it appears that the revenue is being “canceled out” to give a near constant revenue prediction over the course of December. You can see where at the end of the month the revenue is slightly over 16 units. This brings us to our forecast:

A graph showing the difference between the prediction and the prediction

Description automatically generated with medium confidence

Here in our forecast the revenue is rising from 16 units and stabilizes at around 16.24 units. This, from a visual perspective, is clearly a result of the sharp drop off that occurred in the middle of the previous mouth where the model did not seem to capture that extreme fluctuation which is giving us these constant values. More tests will need to be done to say for sure.

**3.** Since our predicted revenue stabilizes to a constant very quickly, it can be assumed that the predictions are not reliable for long term planning, but only adhere to a short-term stabilization. Because of this, I would be very cautious with any immediate decision making. I recommend more tests be done to investigate the integrity of the data, and if the integrity holds true, then I recommend more long-term forecasting to discover any medical trends or patterns that might exist in predicting profits and revenue in the future.

**F. Reporting**

**1.** A report created in jupyter notebook has been attached alongside this written assessment.

**G/H. Sources**

Enemuo, Chukwuebuka, et al. “What Are the Advantages and Disadvantages of Stationarizing Your Data before Forecasting?” Stationarity and Differencing for Time Series Forecasting, www.linkedin.com, 8 Apr. 2024, [www.linkedin.com/advice/0/what-advantages-disadvantages-stationarizing-your-data#:~:text=Differencing%20is%20a%20technique%20to,or%20from%20a%20lagged%20value](http://www.linkedin.com/advice/0/what-advantages-disadvantages-stationarizing-your-data#:~:text=Differencing%20is%20a%20technique%20to,or%20from%20a%20lagged%20value).

Sewell, Dr. William. “D213 ARIMA Webinar I”. 2024. PowerPoint. Western Governors University. <https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william_sewell_wgu_edu/EfYOIw8F-DlBgRcMCunCco4BYfJEjZ-t_CGyV_ujN4-4eQ?rtime=XGDlXLJu3Eg>

Dobbins, Timothy A., et al. “Interrupted Time Series Analysis Using Autoregressive Integrated Moving Average (ARIMA) Models: A Guide for Evaluating Large-Scale Health Interventions - BMC Medical Research Methodology.” SpringerLink, BioMed Central, 22 Mar. 2021, link.springer.com/article/10.1186/s12874-021-01235-8.

Prabhakaran, Selva. “Time Series Analysis in Python - A Comprehensive Guide with Examples - ML+.” Machine Learning Plus, 23 Mar. 2023, www.machinelearningplus.com/time-series/time-series-analysis-python/.