CS215 - Assignment 4

Ramachandran S - 23b1052
Keyaan KR - 23b0977
Harith S - 23b1085

October 26, 2024

Contents

1 Question 1: Parking Lot Problem

1.1 Forecast the total no of vehicles entering the parking lot per day

Table 1: ETS Forecast (Values in Units)

Date	Forecast
2024-11-07	821.25
2024-11-08	824.26
2024-11-09	841.78
2024-11-10	893.59
2024-11-11	853.70
2024-11-12	825.80
2024-11-13	882.31

ETS MASE: 0.636
ETS MAPE: 5.145

1.2 Forecast the average time spent by a vehicle in the parking lot on a day

Table 2: SARIMA Forecast (Values in Units)

Date	Forecast
2024-11-08	340.22
2024-11-09	306.88
2024-11-10	280.87
2024-11-11	271.49
2024-11-12	282.03
2024-11-13	308.66
2024-11-14	341.88

SARIMA MASE: 0.209
SARIMA MAPE: 3.218

1.3 Forecasting the above with outlier smoothening

Table 3: ETS Forecast (Outlier Smoothing: Deletion, Values in Units)

Date	Forecast
2024-11-07	819.23
2024-11-08	815.71
2024-11-09	799.74
2024-11-10	894.52
2024-11-11	848.03
2024-11-12	820.11
2024-11-13	878.41

ETS MASE: 0.554
ETS MAPE: 4.738

Table 4: ETS Forecast (Outlier Smoothing: LOCF, Values in Units)

Date	**Forecast**
2024-11-07	834.45
2024-11-08	828.54
2024-11-09	853.04
2024-11-10	910.81
2024-11-11	857.97
2024-11-12	828.80
2024-11-13	892.75

ETS MASE: 0.570
ETS MAPE: 4.650

Table 5: SARIMA Forecast (Outlier Smoothing: Deletion, Values in Units)

Date	**Forecast**
2024-11-08	345.11
2024-11-09	315.03
2024-11-10	290.67
2024-11-11	279.74
2024-11-12	286.78
2024-11-13	308.71
2024-11-14	337.61

SARIMA MASE: 0.179
SARIMA MAPE: 2.688

Table 6: SARIMA Forecast (Outlier Smoothing: LOCF, Values in Units)

Date	**Forecast**
2024-11-08	347.22
2024-11-09	318.52
2024-11-10	293.96
2024-11-11	282.24
2024-11-12	287.43
2024-11-13	307.55
2024-11-14	335.29

SARIMA MASE: 0.200
SARIMA MAPE: 2.997

2 Question 2: Forecasting on a Real World Dataset

2.1 Sub Part-1

2.1.1 Sub Part-a

Data Cleaning
Firstly, we drop the `AIRLINE` attribute as the number of unique entries in that column are only one. Next, since the values of type float but in the data frame they are of type string. So, we remove the commas and

convert them to floats. Since we also have a few null values as seen in `df.info()`, we fill in the appropriate values by using `ffill()` function.
Since we are doing time series analysis, it would be very helpful to have a date column. Hence we have a mapping between the months and their numbers and we convert to a date accordingly and we sort according to the date and set it as index.

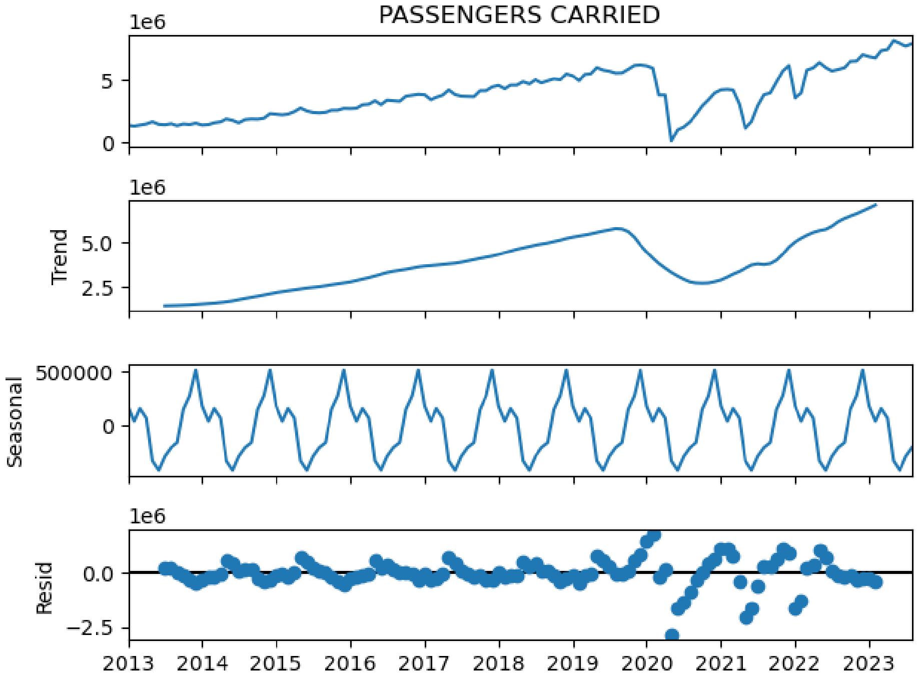

Figure 1: Plots showing trends and seasonality

The huge dip is going to significantly affect our predictions. So, we have to ignore it when are training our data. The plots also show a yearly pattern and the trend seems to be positive if not for the COVID shock. In order to inspect the trends within a year, we look at a plot which shows the passengers carried within an arbitary year.

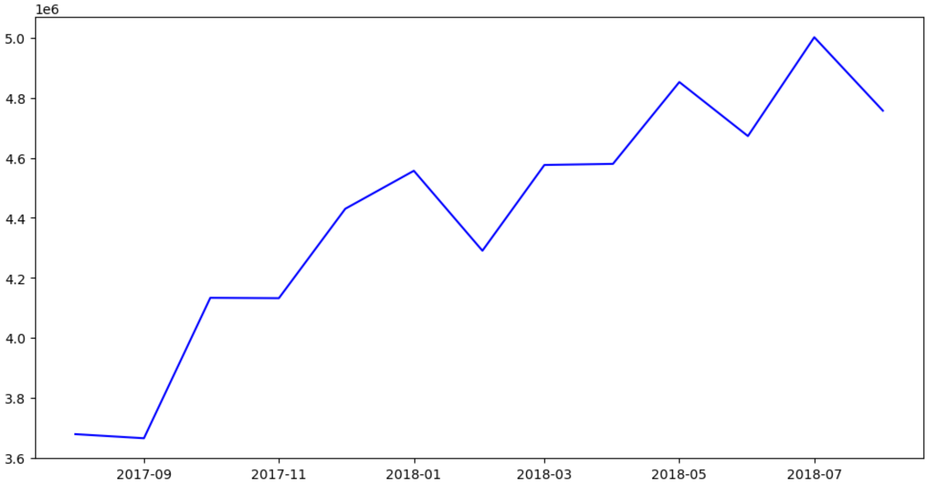

Figure 2: Seasonality within a year

Now, we go on to forecasting. We assume that the change in slope after COVID after the time frame specified goes away, as there should've been this change in slope, but it isn't present in the data given, so it is assumed to be present from SEPT 2023. This means that the slope is same as the one before, and here we can use the help of either using drift or using ARIMA model on pre covid.
Before that though, we interpolate the COVID part so that any irregularities has no effect on the analysis we do. For this, we set all the values between the COVID period to NAN and interpolate it. After this the plot looks like this:

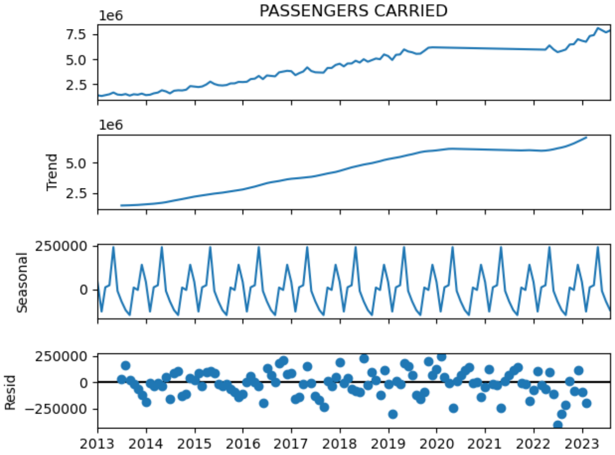

Figure 3: Modified data

Forecasting:
ARIMA:
In order to find the best parameters for ARIMA, we run it over a range, and using the AIC(Akaike Information Criterion) values, we decide the best parameters. Using that we get a plot like this:

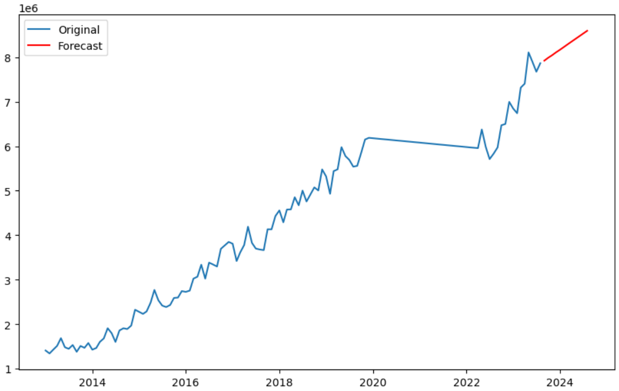

Figure 4: ARIMA

Adding Seasonality:
This is very similar to Naive Drift itself. In order to improve this, we add seasonality into this. We do this we grouping the apssendgers attribute into groups and taking their mean and subtrating the mean of these values so that we know the offsets. We add the offsets to the drift to get a plot as follows:
In an attempt to improve this, SARIMA was used and the seasonalities was taken from the preditictions but that ended up performing worse than this simple method. Since there are a lot irregularities, it is possible that a simple method outperforms slightly complicated methods.

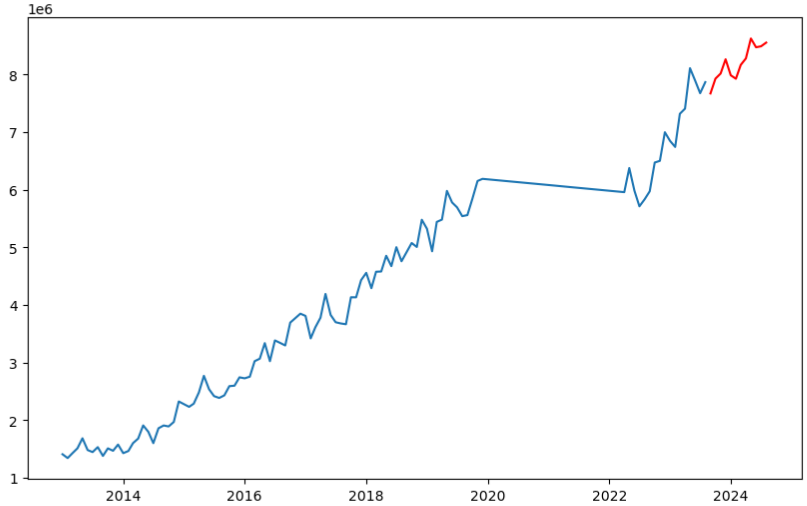

Figure 5: Drift and seasonality

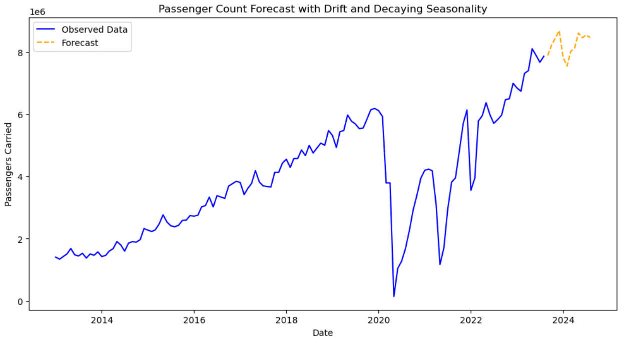

Figure 6: SARIMA seasonalities added

2.1.2 Sub Part-b

The Prompt that was used to generate the prediction was:

Predict the next 12 months of ”Passengers Carried” for a major Indian airline based on the historical monthly data provided from January 2013 to August 2023.
Data Preprocessing Details:
1. Tokenization Strategy: Each digit of the passenger count is separated by a space, and each month’s data point is separated by a comma for optimal tokenization.

Time Series Data (Passengers Carried): 1 4 0 8 0 1 2 , 1 3 4 1 2 1 0 , 1 4 2 3 5 6 9 , 1 5 1 1 0 9 4 , 1 6 8 5 1 6 8 , 1 4 8 0 8 7 9 , 1 4 4 5 2 4 8 , 1 5 3 1 4 0 6 , 1 3 7 8 6 9 1 , 1 5 1 0 1 8 4 , 1 4 6 7 7 6 3 , 1 5 7 5 8 7 2 , 1 4 2 6 5 8 0 , 1 4 6 4 0 7 0 , 1 6 0 1 1 4 1 , 1 6 7 9 9 6 3 , 1 9 0 8 3 3 4 , 1 7 9 7 1 0 1 , 1 5 9 9 9 7 7 , 1 8 5 8 6 6 4 , 1 9 0 7 3 7 8 , 1 8 9 0 2 7 3 , 1 9 6 7 9 9 2 , 2 3 2 4 2 2 1 , 2 2 7 6 4 0 4 , 2 2 3 0 6 4 5 , 2 2 8 6 1 2 8 , 2 4 8 1 2 8 5 , 2 7 6 9 2 8 3 , 2 5 3 6 5 5 4 , 2 4 1 6 9 1 6 , 2 3 8 4 9 4 3 , 2 4 3 0 4 4 9 , 2 5 8 9 8 6 1 , 2 5 9 7 7 6 5 , 2 7 4 3 3 2 5 , 2 7 2 5 7 1 1 , 2 7 5 4 1 3 1 , 3 0 2 3 2 2 8 , 3 0 6 6 5 5 6 , 3 3 3 6 8 3 9 , 3 0 2 3 0 8 1 , 3 3 8 3 7 6 8 , 3 3 4 1 0 8 1 , 3 2 9 5 8 2 6 , 3 6 9 2 8 2 8 , 3 7 7 2 5 8 3 , 3 8 4 8 3 2 2 , 3 8 0 9 2 2 8 , 3 4 1 8 6 0 5 , 3 6 1 1 3 7 1 , 3 7 7 8 7 8 0 , 4 1 9 0 9 1 4 , 3 8 2 5 8 1 4 , 3 6 9 9 4 5 1 , 3 6 7 8 2 4 5 , 3 6 6 4 5 0 9 , 4 1 3 3 0 2 7 , 4 1 3 1 8 4 4 , 4 4 3 0 0 7 0 , 4 5 5 6 9 0 4 , 4 2 9 0 1 8 9 , 4 5 7 6 2 3 6 , 4 5 7 9 9 1 6 , 4 8 5 2 9 0 9 , 4 6 7 2 6 8 6 , 5 0 0 2 4 1 6 , 4 7 5 7 3 7 8 , 4 9 2 0 3 3 5 , 5 0 7 4 8 5 3 , 5 0 0 5 9 1 9 , 5 4 7 8 5 2 3 , 5 3 2 1 8 3 2 , 4 9 3 0 6 0 8 , 5 4 4 0 7 9 6 , 5 4 8 1 0 8 8 , 5 9 7 9 5 5 1

, 5 7 7 8 3 7 6 , 5 6 9 3 4 4 8 , 5 5 4 0 6 9 3 , 5 5 5 9 0 3 1 , 5 8 4 2 3 1 8 , 6 1 5 1 3 4 7 , 6 1 8 8 4 5 7 ,
6 1 1 9 9 2 4 , 5 9 3 1 1 8 9 , 3 7 9 3 4 6 4 , N U L L , 1 4 2 2 5 4 , 1 0 4 6 8 3 6 , 1 2 7 2 2 4 0 , 1 6 8 1
7 8 7 , 2 2 6 5 5 3 9 , 2 9 2 6 8 3 5 , 3 4 2 3 0 5 9 , 3 9 5 1 5 2 2 , 4 2 0 2 7 1 2 , 4 2 3 8 4 6 6 , 4 1 8 4 9
2 0 , 3 0 8 3 3 1 8 , 1 1 6 8 7 3 8 , 1 7 0 1 9 5 5 , 2 9 3 2 4 6 9 , 3 8 1 6 2 8 6 , 3 9 5 8 3 7 5 , 4 8 0 7 4 1
5 , 5 7 0 6 0 2 1 , 6 1 4 0 6 5 4 , 3 5 5 6 5 5 8 , 3 9 5 0 5 9 7 , 5 7 8 3 8 3 8 , 5 9 5 7 4 1 1 , 6 3 7 6 4 2 9
, 5 9 8 2 7 8 7 , 5 7 1 1 2 8 8 , 5 8 3 2 0 2 5 , 5 9 7 2 4 4 9 , 6 4 7 1 4 4 1 , 6 5 0 0 9 0 3 , 6 9 9 7 4 7 2 ,
6 8 4 7 3 8 4 , 6 7 4 1 9 4 8 , 7 3 1 7 2 8 8 , 7 4 0 6 4 4 0 , 8 1 0 9 6 2 6 , 7 8 9 3 2 9 6 , 7 6 7 4 8 9 0 , 7 8 6 7 2 5 9

Time Series Model: Train a SARIMA model using the first 86 values and use the model to predict what might be the next 24 values. Now using the 128 values (86 given in input, followed by the 24 values created using prediction and last 18 values from the input), predict the next 12 values. Note: There could be a seasonality of 12 because we are working the data of passenger travelled from an airline in every month.
Using this format, please predict "Passengers Carried" for the following 12 months from 2023 Sep to 2024 Aug.

Ensure that:
- The output is structured similarly, with each month's forecasted value separated by a comma.
- Each digit of the passenger count remains separated by spaces to align with the tokenization approach.
- Report the predictions in the format below:

Output Format: # # # # # , # # # # # , ..., # # # # #

The Evaluation for the output was done using MAPE and the evaluation gave **4.34% MAPE**.

2.1.3 Sub Part-c

A Global model was trained using the "Prophet" Library and once the model was fit using the given time series data, it was used to predict the values of "PASSENGERS CARRIED" for the next 12 months. The Predicted values are printed in the python notebook and their corresponding evaluation based on MAPE is **3.18%**

2.2 Sub Part-2

2.2.1 Fleet Management

Fleet management relies on accurately forecasting the total passenger count over a quarter or longer to make informed decisions about resource allocation. The goal is to ensure the fleet size can comfortably meet the expected demand without under- or overestimating capacity requirements.

Why MAPE Is Not Ideal for Fleet Management:

- **Misleading Representation of Total Demand:** MAPE's percentage-based errors do not reflect the absolute number of passengers by which the forecast deviates. Fleet planning requires an understanding of absolute discrepancies to assess how many more or fewer seats or planes might be needed. MAPE can obscure these absolute errors, leading to potentially inefficient resource allocation.
- **Undervaluing High-Demand Periods:** Since MAPE gives equal weight to errors across all periods, it doesn't prioritize periods where fleet utilization is maximized. Fleet management decisions are more sensitive to total demand across all months, and MAPE's emphasis on percentage deviations can undervalue errors in high-demand periods that drive overall capacity needs.

Example: Imagine a month with a low demand forecast (e.g., 1,000 passengers) and an actual deviation of 100 passengers (10% MAPE). MAPE would mark this as a significant error, even though, in terms of fleet management, an extra 100 seats may not be critical. In contrast, during a peak month with 10,000 passengers, a deviation of 500 passengers (just 5% MAPE) would imply a significant underestimation of needed fleet capacity. For fleet management, the absolute count error (MAE) better aligns with planning needs by providing a clearer view of total discrepancy.

MAE (Mean Absolute Error) for Fleet Management:

- **Why MAE?** MAE calculates the average absolute deviation of the forecasted values from the actual values, expressed in the same units (passengers in this case). It directly measures the magnitude of errors without considering the relative size of the actual values, making it straightforward to understand the typical error in total passenger counts.
- **Example:** If the forecasted total number of passengers is consistently off by 500 on average, MAE would clearly indicate this deviation without exaggerating the error during low-demand months or understating it during high-demand months.

2.2.2 Human Resources

Human resources planning must focus on peak demand periods, as these are when staffing shortages are most likely to affect service quality. Unlike fleet management, which cares about aggregate demand, HR requires precise forecasting during high-demand times to avoid under-staffing during peaks and over-staffing during troughs.

Why MAPE Is Not Ideal for Human Resources Planning:

- **Equal Weight to All Periods:** MAPE gives equal importance to all data points, meaning errors during peak periods (when accurate forecasting is most critical) are not distinguished from errors in low-demand periods. Staffing needs are driven by peak times, so a metric that amplifies errors during high-demand months is better suited for HR planning.
- **Potential Misalignment with Actual Staffing Needs:** Since MAPE is insensitive to the magnitude of errors in absolute terms, it may indicate high accuracy overall even if peak forecasts are off, which could be critical for HR. This leads to the risk of under-staffing during peak periods if MAPE suggests satisfactory forecast accuracy, while in reality, high-demand periods are inaccurately predicted.

Example: In a low-demand period with only 500 passengers, an error of 50 passengers leads to a 10% MAPE. In a peak period with 5,000 passengers, a 250-passenger error leads to a 5% MAPE, suggesting that the low-demand period is "worse" in terms of accuracy. For HR, however, the 250-passenger error in the peak period would be far more critical for scheduling staff to meet demand.

RMSE (Root Mean Square Error) or Max Error for Peak Demand (Human Resources):

- **Why RMSE?** RMSE gives higher weight to larger errors because it squares each deviation before averaging. This feature makes RMSE more sensitive to periods with high errors, which is beneficial for forecasting peak demands where accurate predictions are critical.
- **Max Error** could be an alternative or supplement to RMSE, as it reports the single largest forecast error, which is often essential in identifying the maximum staffing requirement.
- **Example:** If the forecast for a peak month is off by 2,000 passengers, this large error will significantly increase RMSE, drawing attention to the need for more accurate forecasting in peak months to avoid potential staffing shortages.

2.3 Sub Part-3

Two-Sample t-Test

A two-sample t-test is a statistical test used to determine if there is a significant difference between the means of two independent groups. It's commonly applied when you want to compare the average values (means) of a variable across two different conditions or time periods.

In this case, the two groups are:

1. **Pre-COVID period**: Data from before December 2019.

2. **Post-COVID period**: Data from after January 2022.

The test checks if the mean of the differenced series (μ) in the pre-COVID period is significantly different from the mean in the post-COVID period.

Test Requirements and Why It's Suitable Here
The two-sample t-test assumes that:

- The samples are independent of each other.
- The data in each group is approximately normally distributed (or the sample size is large enough for the Central Limit Theorem to apply).
- The variances of the two groups are roughly equal, although versions of the test (like Welch's t-test) handle cases with unequal variances.

Since we're told that the differenced series (ΔY) is weakly stationary and normally distributed (with $\Delta Y = \mu + N(0, \sigma)$), we meet the requirements of normality and constant variance. These properties make the two-sample t-test an appropriate choice.