2024

Passenger’s Flight Model

FORECASTING THE DEMAND FOR FLIGHT TICKETS OF AN AIRLINE.

OLAWALE CHARLES OLUWOLE, 35791522

Background and goal of the project

For airlines to make a profit, they must know when to raise or reduce ticket prices. Inorder to predict future demands for airline tickets, I developed a model to forecast the demands of flight tickets using an American airline. Additionally, We explored the trends, additive and multiplicative time series decomposition in our analysis.

Description of the data

- The dataset is a standard univariate time series with the time in months from January 1949 to December of 1960,
- The number of passengers from that airline in thousands in each respective month.
- The set had no missing values and no outliers were preprocessed.
- The data source is

  https://www.kaggle.com/datasets/andreazzini/international-airline-passengers

```
[14]: #
      df1.plot(grid=True)
t[14]: <Axes: xlabel='Month'>
```

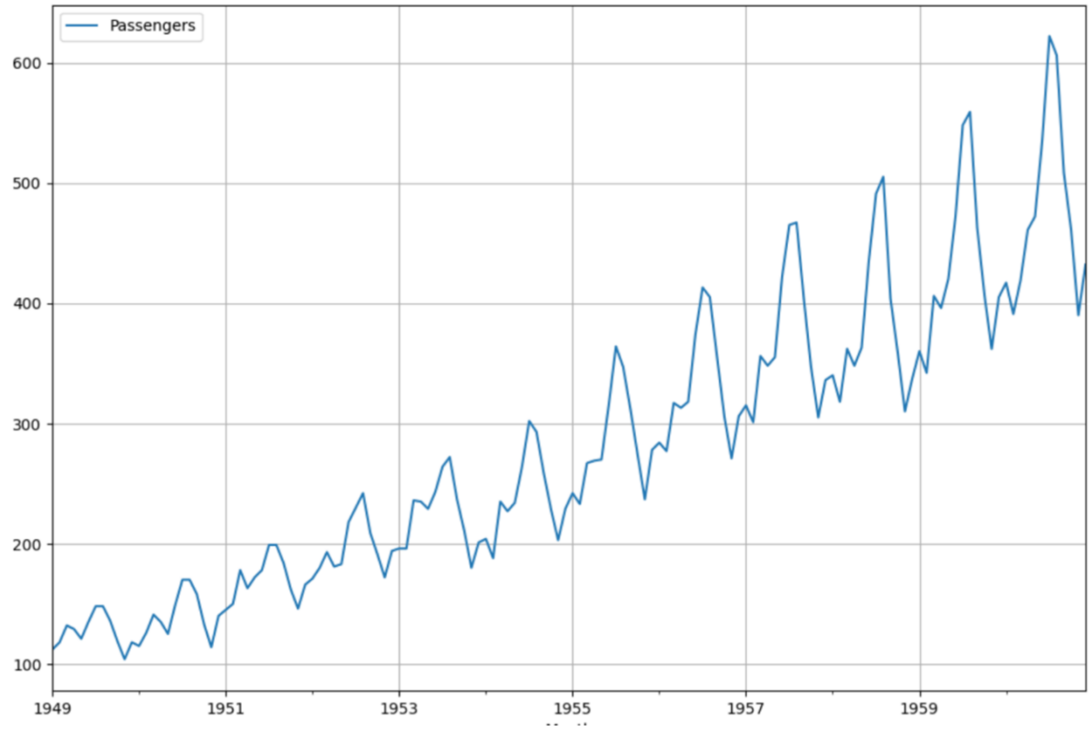

Fig.1

I observe in fig.1 that dataset contains both a trend and seasonality, also as years goes up the number of passengers increases, I could see peaks which could subject to trends. I am going to build a forecasting model to predict both in-sample and out-sample forecast.

```
5]: #Checking for outliers in our dataset
    import seaborn as sns
    sns.violinplot(x=df1['Passengers'])

5]: <AxesSubplot:xlabel='Passengers'>
```

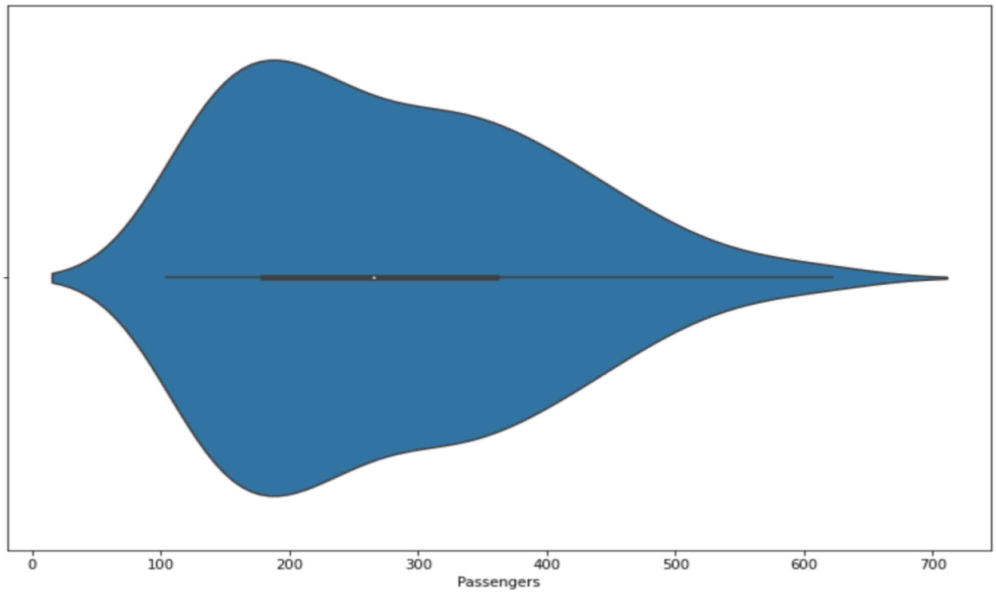

Fig.2

In Fig.2 above, I try to check if I have an outliers in my dataset, and from the plot no outliers were detected.

Make an in-sample forecast

I generated predictions based on the input data used for training the model. Ideally, if the model has encountered the data during training, it should yield accurate predictions. However, in practice, the model attempts to generalize across all instances in the data, leading to imperfect predictions. This process is referred to as making an in-sample forecast (within the training set), and analyzing the outcomes provides valuable insights into the model's performance essentially, how effectively it has grasped the patterns within the training data.

```
In [29]: import seaborn as sns
         sns.lineplot(data=new)

Out[29]: <AxesSubplot:>
```

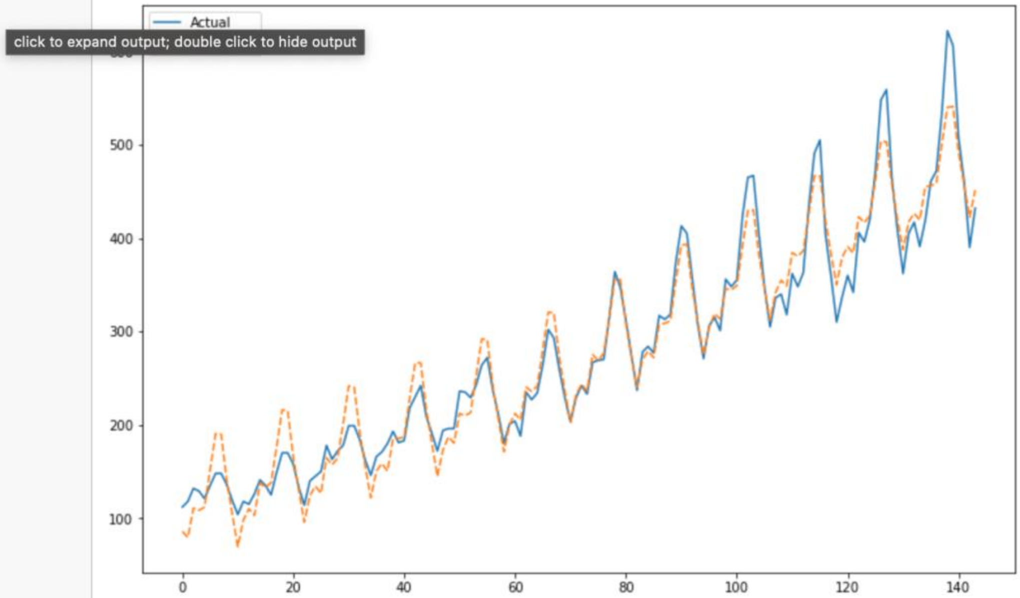

Fig.3

Fig.3 clearly shows that our model captures the general trend of our training data quite well. Besides that, I observed fluctuations in passenger numbers which are caused by the yearly trend.

The Mean absolute error MAE=17.336 implies that, on average, the forecast's distance from the true value is 17.336, it also implies that, on average, the forecast's distance from the true value is 17% of the true value.

Make an Out-sample forecast

Prediction was made based on data outside of the inputted data. This is necessary for assessing how well the model generalises to new, unseen and evaluating its predictive accuracy. I use the “.make_future_future_dataframe(periods = 365)” in the prophet model to generate a new dataframe for future 365 days periods, and also populate it using “.Predict”.

```
[38]: # Python
forecast_plot = model_prophet.plot(forecast)
plt.legend(['Actual', 'Trend', 'Seasonal'])
plt.show()
```

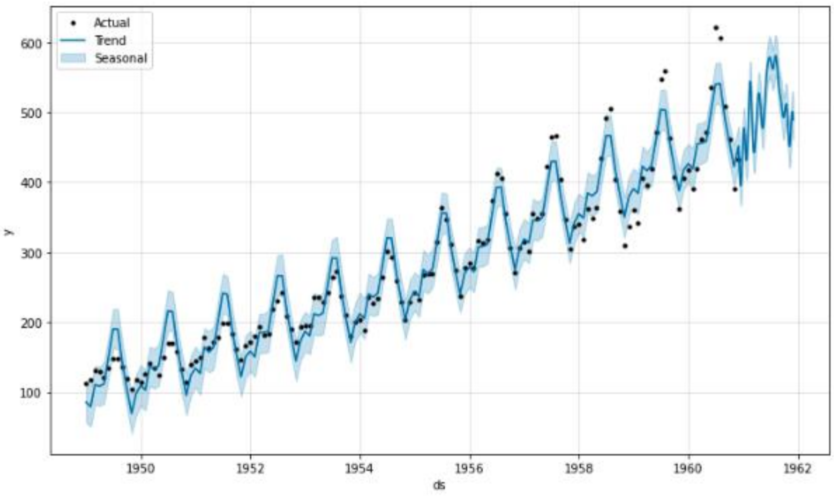

Fig.4

Fig.4 captures general trends similar to our training data, and can also see up and down movement of number of passengers.

Results

```
[31]: components = model_prophet.plot_components(train_forecast)
```

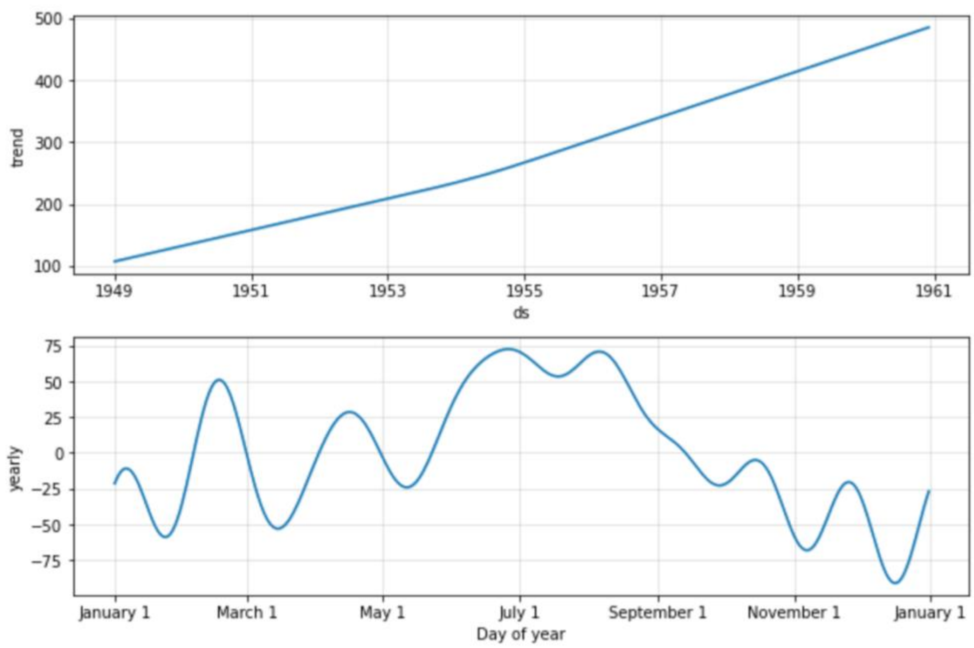

Fig.5

```
[39]: #
forecast_components = model_prophet.plot_components(forecast)
```

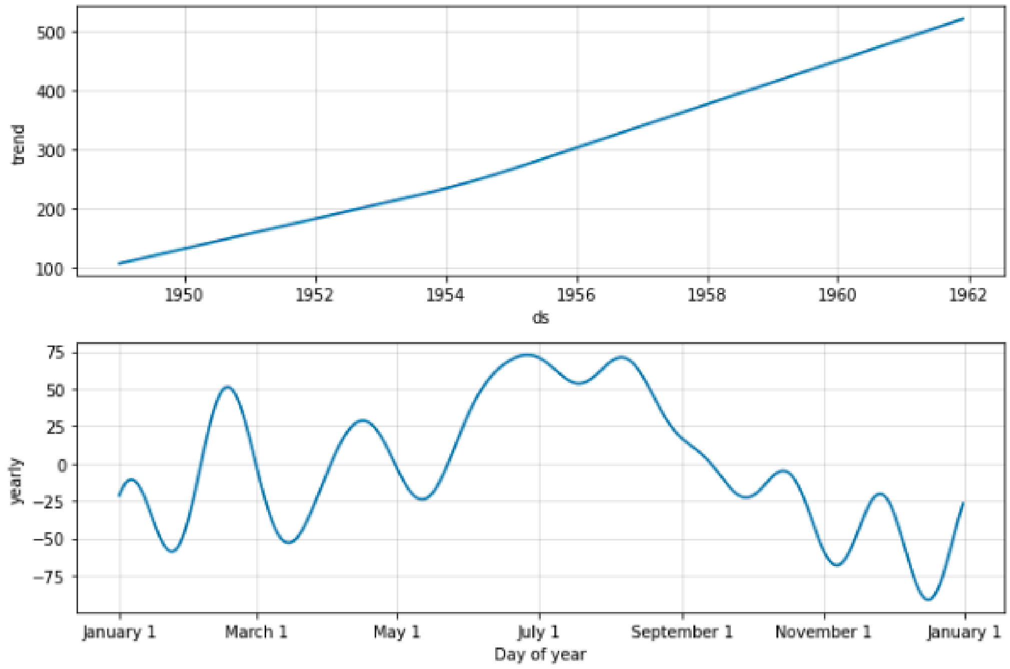

Fig.6

Fig.5 and Fig.6 captures the ouput of in-sample and out-sample forecast respectively. The month of July from the bove plots have the highest numbers of passengers all through the years, and January been lowest based on number of passengers. From the plots, the airline can know when to increase or lower the price of airline tickets to obtain a good margin.

Forecast model evaluation

I created a test data set by droping the last 12 months in our train data set in order to let our model predict these months. This is necessary for comparing the months (1960) in my train data against my test data. The mean absolute error turned out to be 33.45.

```
In [49]: # plot expected vs actual
         pyplot.plot(y_true_eval, label='Actual')
         pyplot.plot(y_pred_eval, label='Predicted')
         pyplot.legend()
         pyplot.show()
```

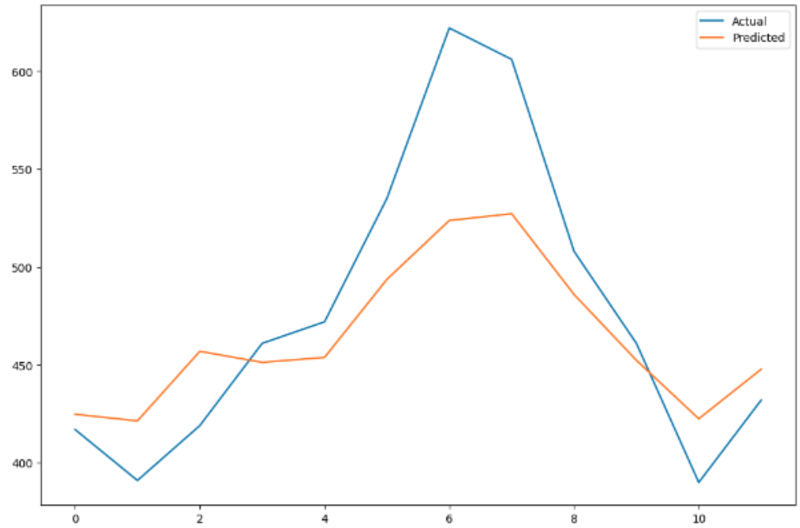

Fig.7

In Fig.7, our plot captured the difference between actual data and our predicted for the months in 1960.

In [15]:

```python
import statsmodels.api as sm
decomposition = sm.tsa.seasonal_decompose(df1, model='additive')
decomposition.plot()
```

Out[15]:

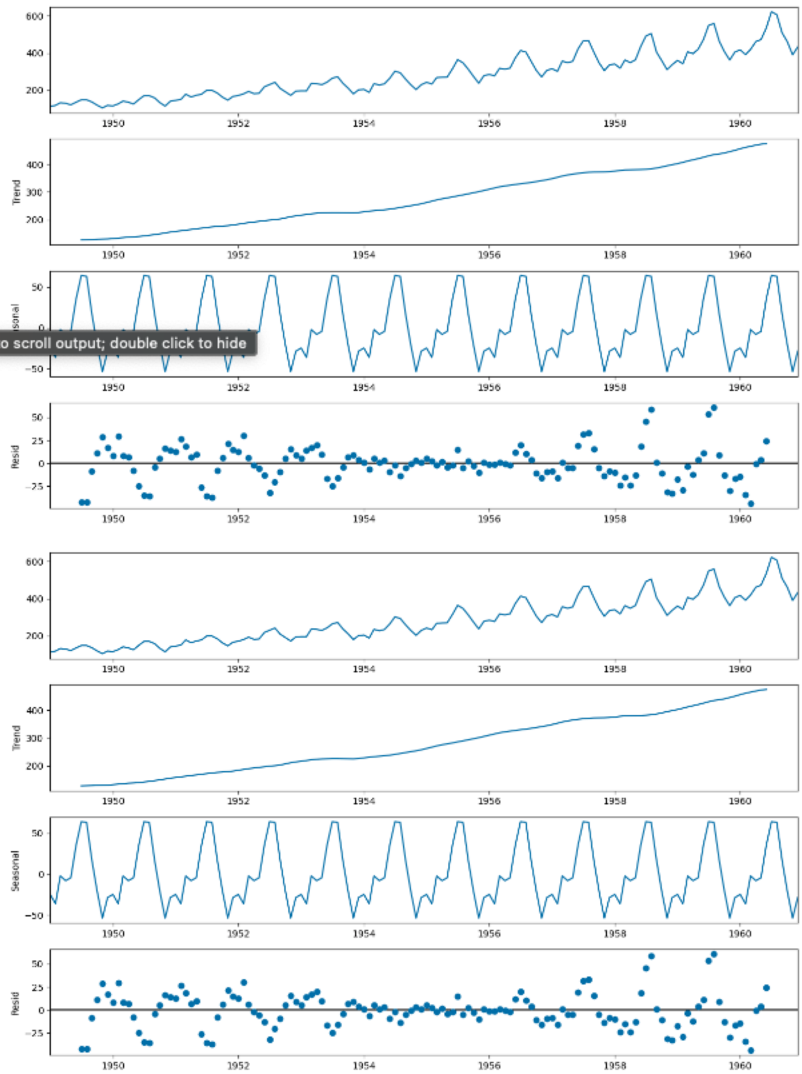

Fig.8

In [16]:
```
decomposition = sm.tsa.seasonal_decompose(df1, model= 'multiplicative')
decomposition.plot()
```

Out[16]:

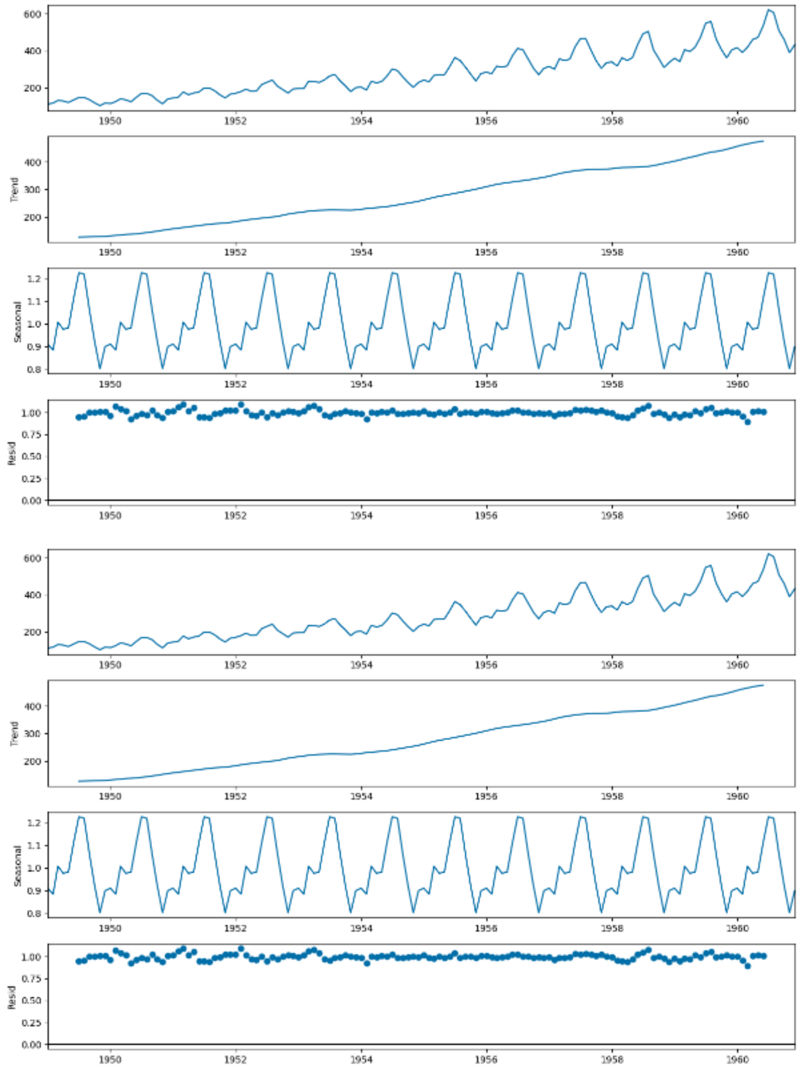

Fig.9

In the Fig.8 and Fig.9, I decomposed the data using the additive model, and the multiplicative model respectively. Both decompositions are very similar with respect to their seasonality. The only difference is the residual.

References

https://www.kaggle.com/datasets/andreazzini/international-airline-passengers