Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)

Report on Mini Project

Time Series Analysis (DJ19DSC5012) AY: 2021-22

Name : Sowmya Dadheech SAP ID : 60009210163

Name : Virum Ranka SAP ID : 60009210165

Name : Dwisha Shah SAP ID : 60009210174

Guided By : Prof. Shruti Mathur

TABLE OF CONTENTS

INTRODUCTION

Given the variations in the Air Traffic at airport, we chose the dataset of Air Traffic at SanFrancisco Airport (SFO) from the year 2005 - 2018.

To study these variations , irregularities in the passenger count, correct time series model(s) and techniques must be used.

The report comprehensively covers the entire workflow, from data cleaning and preprocessing to model fitting and forecasting the possible Air Traffic values based on the historical data.

DATA DESCRIPTION

The Dataset contains of 12 columns. Some columns within the dataset were non-essential to the prediction process and were consequently excluded from consideration.
Below are the columns that were considered :

- **Activity Period** → The year and month at which passenger, cargo or landings activity took place.
- **Operating Airline** → Airline name for the operator of aircraft with passenger, cargo or landings activity.
- **Published Airline** → Airline name that issues the ticket and books revenue for passenger, cargo or landings activity.
- **GEO Summary** → Designates whether the passenger, cargo or landings activity in relation to SFO arrived from or departed to a location within the United States (“domestic”), or outside the United States (“international”) without stops.
- **GEO Region** → Provides a more detailed breakdown of the GEO Summary field to designate the region in the world where activity in relation to SFO arrived from or departed to without stops.
- **Passenger Count** → No. of passengers travelled in a specific period.

	Activity Period	Operating Airline	Operating Airline IATA Code	Published Airline	Published Airline IATA Code	GEO Summary	GEO Region	Activity Type Code	Price Category Code	Terminal	Boarding Area	Passenger Count
0	200507	ATA Airlines	TZ	ATA Airlines	TZ	Domestic	US	Deplaned	Low Fare	Terminal 1	B	27271
1	200507	ATA Airlines	TZ	ATA Airlines	TZ	Domestic	US	Enplaned	Low Fare	Terminal 1	B	29131
2	200507	ATA Airlines	TZ	ATA Airlines	TZ	Domestic	US	Thru / Transit	Low Fare	Terminal 1	B	5415
3	200507	Air Canada	AC	Air Canada	AC	International	Canada	Deplaned	Other	Terminal 1	B	35156
4	200507	Air Canada	AC	Air Canada	AC	International	Canada	Enplaned	Other	Terminal 1	B	34090

Columns not required are dropped from Dataframe.

OBJECTIVE

Objective of the above Time series data includes :

1. Data analysis and cleaning
2. Finding out trends in data
3. Checking for Seasonality
4. Smoothening the data
5. Identifying parameters of the model
6. Fitting the right model
7. Calculating accuracy matrix
8. Forecasting future prices

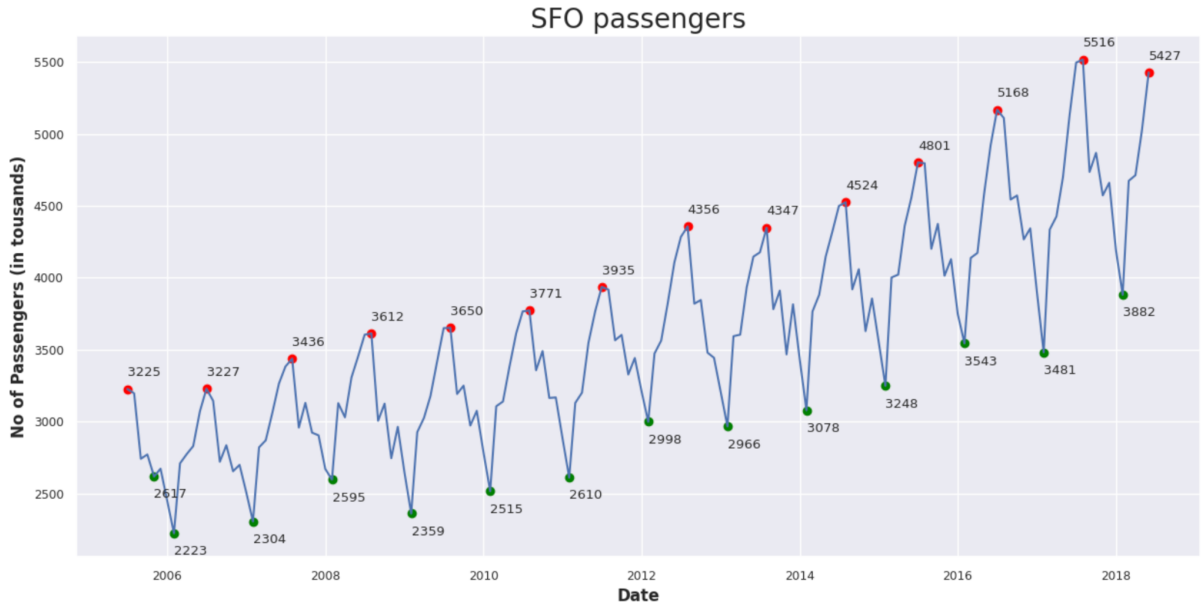
Data Analysis and Cleaning

The data contains 18885 rows and 12 feature columns.

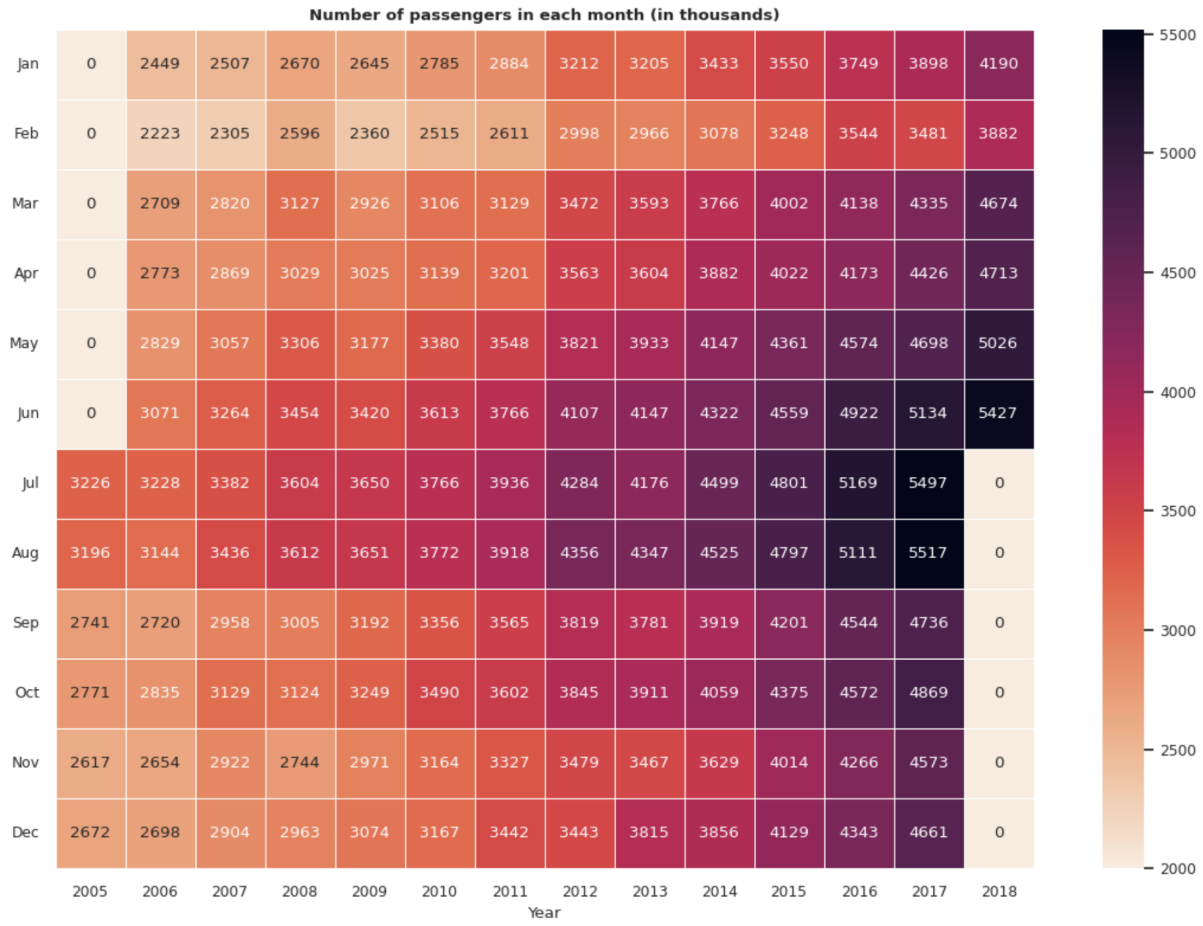
Only 2 columns contain missing information (i.e Operating Airline IATA Code, Published Airline IATA Code). However, these columns are not relevant for this analysis as these are just IATA Airline Codes.

For better pre-processing, we have formatted the `Activity Period` to a date type and extract year and month components.

The data is reported the time range First date: 2005-07-01 and
Last date: 2018-06-01.



The above graph shows the overall number of passengers at San Francisco airport over time. The number of passengers is generally increasing (trend) with a clear seasonal pattern with a period of about one year. The traffic peak is around summertime while the lowest traffic is during wintertime at the beginning of the year.



The heatmap above shows the number of passengers in each month for years 2005-2018 (in thousands). We can see a peak in traffic around June-August. Such a peak makes sense as this is a typical holiday period in USA.

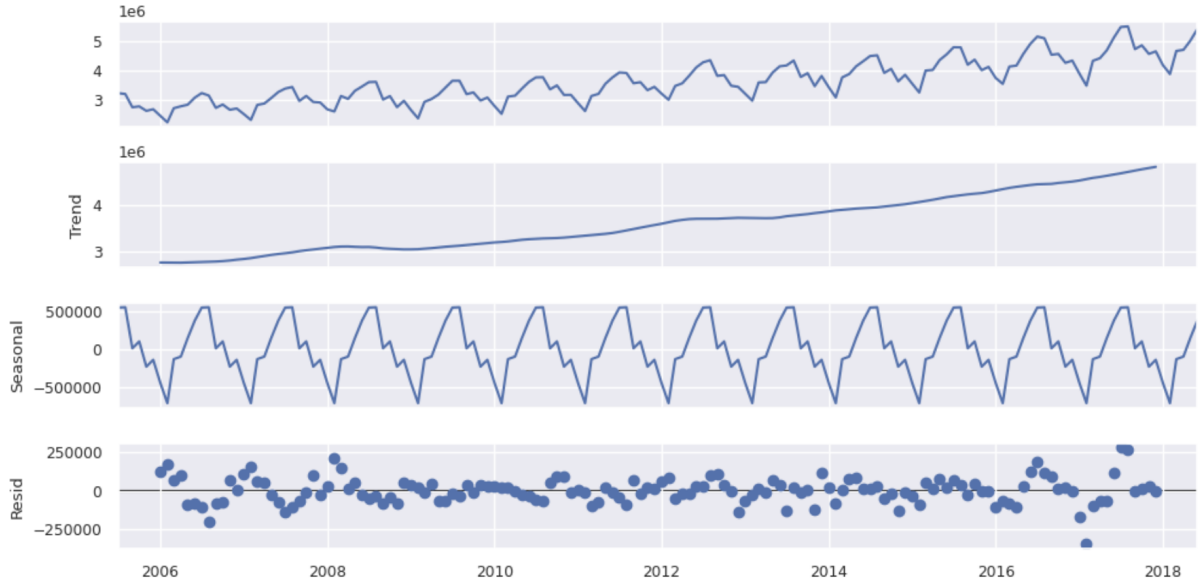
Asia and Europe are the geographic regions having the highest share in traffic generated.

DATA DECOMPOSITION

Any Time Series data consists of 3 components :

1. **Trend**
2. **Seasonal**
3. **Residual**

This helps in better understanding the time series and in Identifying seasonality & trend if present.



The model seems to have a 12-month seasonal trend which makes sense as usually every year a holiday period is more busy at airports.

TESTING FOR STATIONARITY

The Augmented Dickey-Fuller (ADF) test can be used to test for stationarity of a time series. Checking for unit roots is mandatory because not all models can be applied to a TS that contains unit roots.

```
def Augmented_Dickey_Fuller_Test_func(series, column_name):
    print (f'Results of Dickey-Fuller Test for column: {column_name}')
    dftest = adfuller(series, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','No Lags Used','Number of Observations Used'])
    for key,value in dftest[4].items():
       dfoutput['Critical Value (%s)'%key] = value
    print (dfoutput)
    if dftest[1] <= 0.05:
        print("Conclusion:====>")
        print("Reject the null hypothesis")
        print("Data is stationary")
    else:
        print("Conclusion:====>")
        print("Fail to reject the null hypothesis")
        print("Data is non-stationary")
```

The below results show that the data is non-stationary.

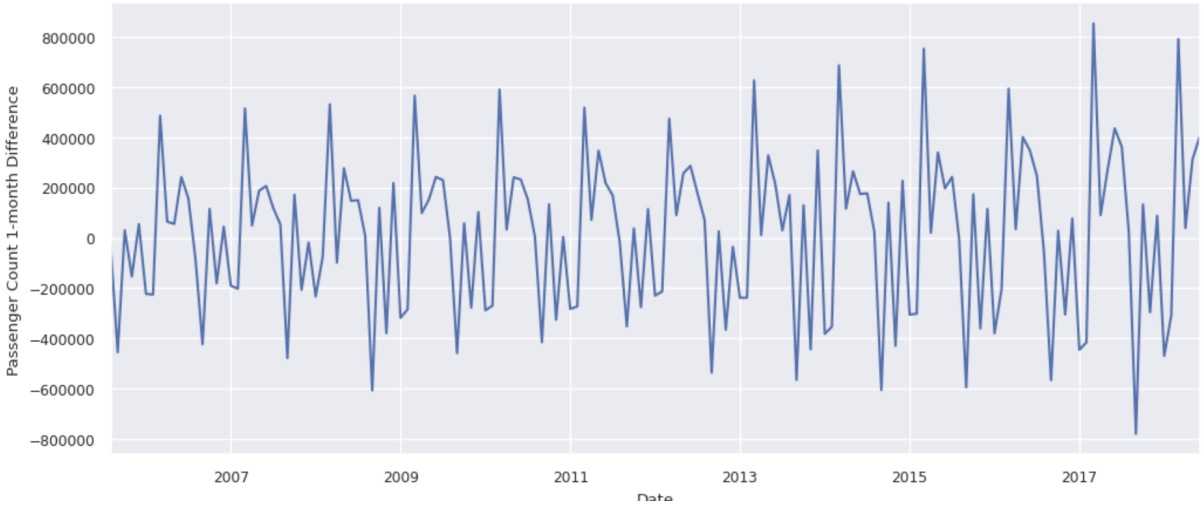
```
Augmented_Dickey_Fuller_Test_func(TS1["Passenger Count"],'Passenger Count')
```

```
Results of Dickey-Fuller Test for column: Passenger Count
Test Statistic                  1.133106
p-value                         0.995489
No Lags Used                   14.000000
Number of Observations Used    141.000000
Critical Value (1%)             -3.477601
Critical Value (5%)             -2.882266
Critical Value (10%)            -2.577822
dtype: float64
Conclusion:====>
Fail to reject the null hypothesis
Data is non-stationary
```

We accept the null hypothesis since **p-value > 0.05 and test-stat > critical values**.

In order to stationarize time series, apply differencing step (1 step = 1 month).

```
TS1_diff = TS1.diff().dropna()

plt.figure(figsize=(12,5))
ax = TS1_diff["Passenger Count"].plot()
ax.set_xlabel("Date")
ax.set_ylabel("Passenger Count 1-month Difference")
plt.grid(True)
plt.show()
```



Checking for stationarity again using ADF test.

```
Results of Dickey-Fuller Test for column: Passenger Count
Test Statistic                  -2.950493
p-value                          0.039784
No Lags Used                    13.000000
Number of Observations Used    141.000000
Critical Value (1%)             -3.477601
Critical Value (5%)             -2.882266
Critical Value (10%)            -2.577822
dtype: float64
Conclusion:====>
Reject the null hypothesis
Data is stationary
```

The data becomes stationary after applying 1 differencing.

WHY? Justification of TS

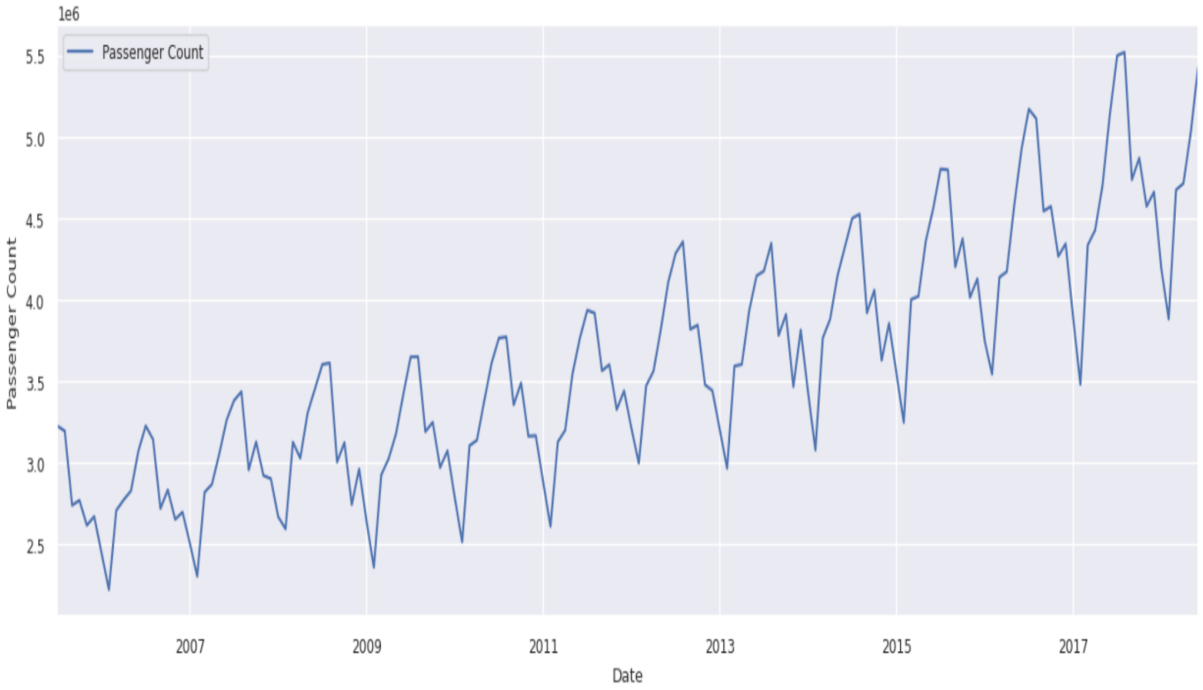
Reasons for classifying the data as a TS data :

1. The number of passengers arriving at SFO Airport is continuously increasing with some irregularities.

2. The data is continuous i.e. there exists rate for every timestamp.

3. There is a seasonality, over the period of 1 year.

4. This data can be decomposed into the three components of TS.



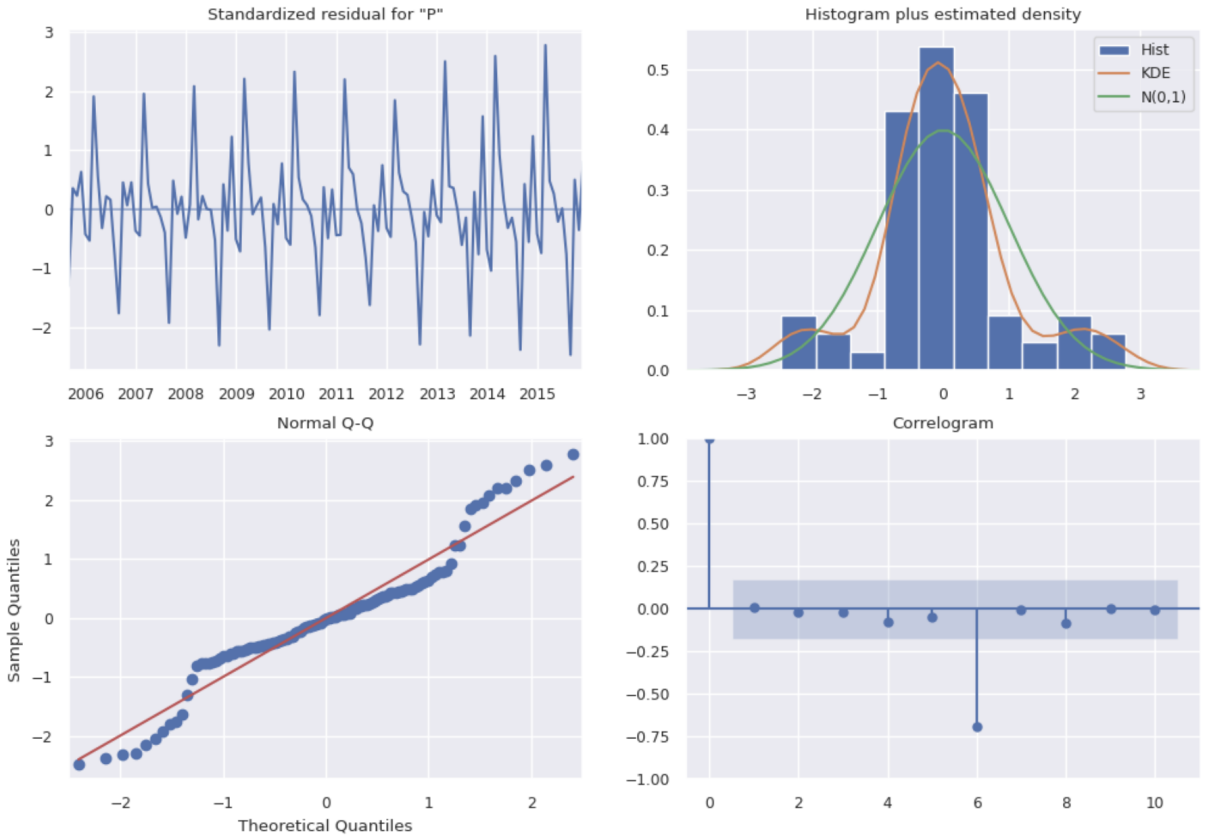
IMPLEMENTATION AND FORECASTING

Various methods and algorithms can be applied for implementing a time series model and interpretation of its forecast generated.
From implementation we can interpret that forecasts generated by models are able to capture irregularity in data and give somewhat accurate predictions.
Some methods implemented by us are :

- **Autoregressive Integrated Moving Average(ARIMA) Model**

  Autoregressive integrated moving average—also called ARIMA(p,d,q)—is a forecasting equation that can make time series stationary with the help of differencing and log techniques when required. A time series that should be differentiated to be stationary is an integrated (d) (I) series. Lags of the stationary series are classified as autoregressive (p), which is designated in (AR) terms. Lags of the forecast errors are classified as moving averages (q), which are identified in (MA) terms.

```
from statsmodels.tsa.statespace.sarimax import SARIMAX

TS1.index.freq = TS1.index.inferred_freq
ARIMA = SARIMAX(train, order=(2,1,1))
result = ARIMA.fit()
result.summary()
```



- **Convolutional Neural Network(CNN) Model**

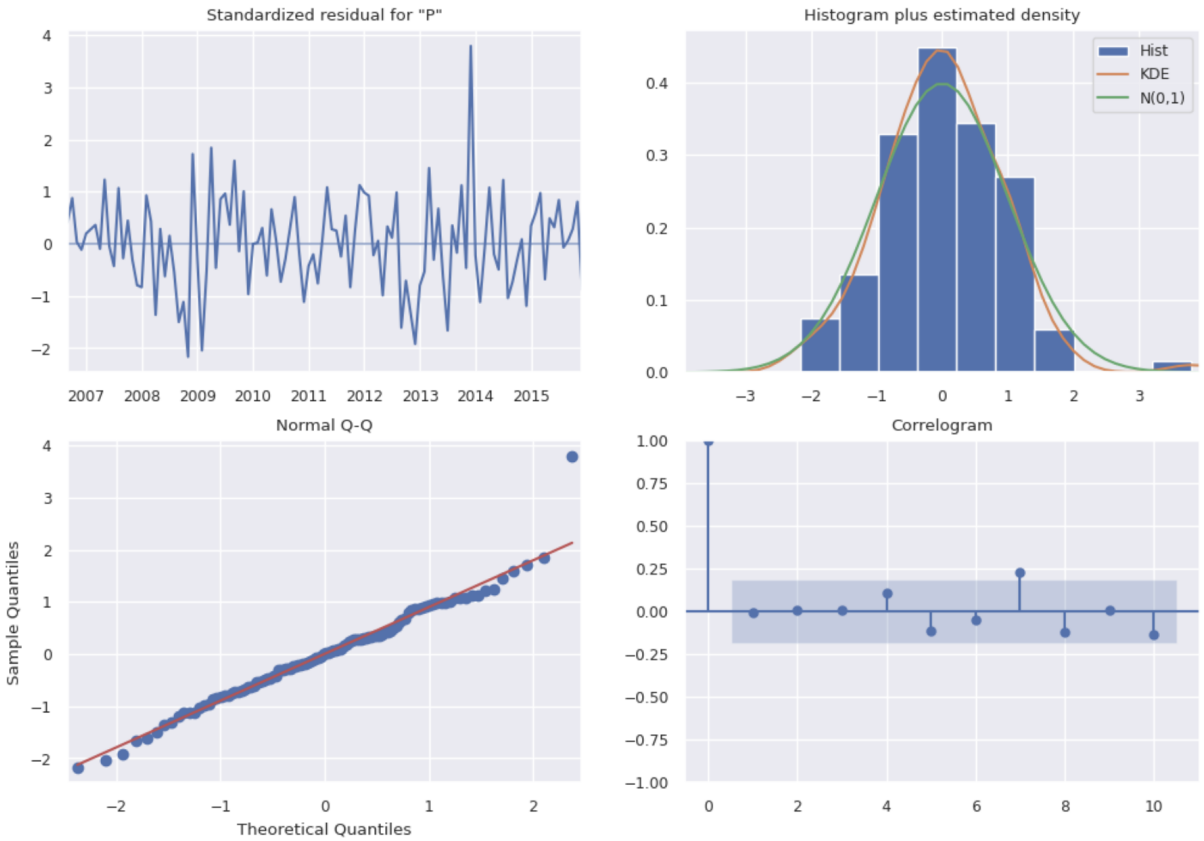
  We have seen examples on using CNN for sequence prediction. If we consider Dow Jones Industrial Average (DJIA) as an example, we may build a CNN with 1D convolution for prediction. This makes sense because a 1D convolution on a time series is roughly computing its moving average or using digital signal processing terms, applying a filter to the time series. It should provide some clues about the trend.

```
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=2, activation='relu', input_shape=(steps, features)))
model.add(MaxPooling1D(pool_size=2))

model.add(Flatten())
model.add(Dense(100, activation="relu"))
model.add(Dense(1))
```

- **SARIMA (Seasonal Auto-Regressive Integrated Moving Average)**

  It is an extension of the ARIMA model that incorporates ***seasonality*** in addition to the non-seasonal components. ARIMA models are widely used for time series analysis and forecasting, while SARIMA models are specifically designed to handle data with seasonal patterns.

```
from statsmodels.tsa.statespace.sarimax import SARIMAX

TS1.index.freq = TS1.index.inferred_freq
SARIMA = SARIMAX(train, order=(2,1,1), seasonal_order=(0,1,0,12))
result1 = SARIMA.fit()
result1.summary()
```

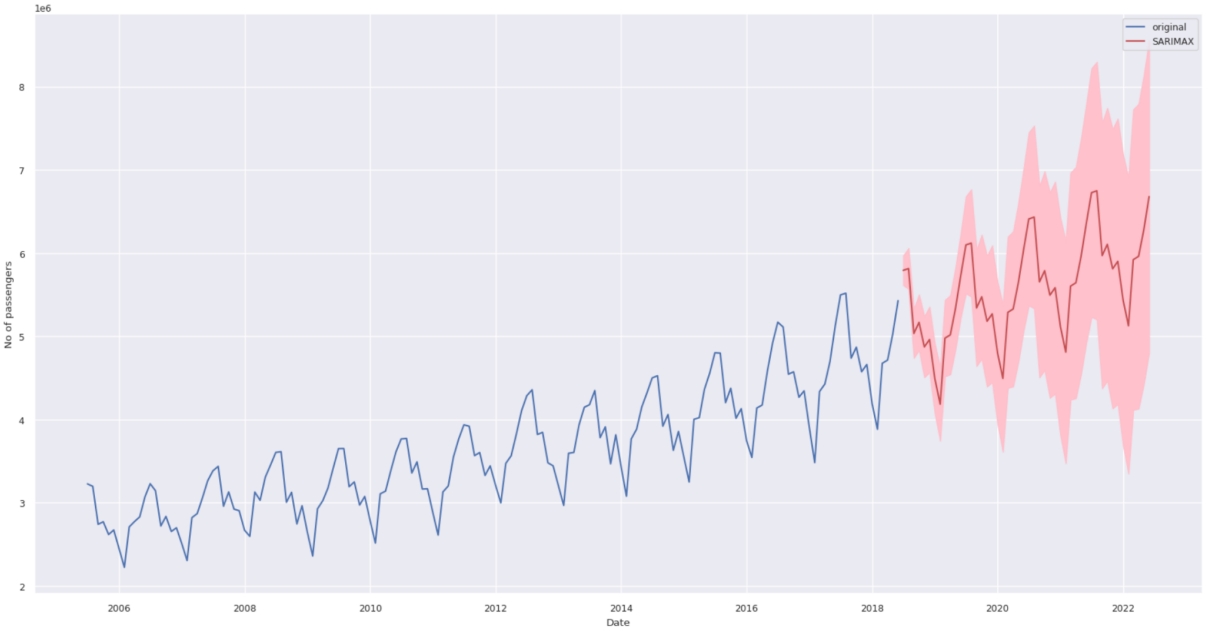


- **OPTIMIZATION**

Splitting into train and test data the models were not giving better results as the important features of the data were separated which couldn't capture the trends and seasonality efficiently resulting in poor future predictions.

Before:

After :



SELECTION OF MODEL

For this TS data , **SARIMA** model was used , based on various metrics like mse, rmse, R2, etc.

Reasons for SARIMA model :

- The Air Traffic data was not very volatile, since the use of **ARCH/GARCH** model.
- Data was made stationary (No Unit roots). Thus an **AR/MA** or ARIMA or CNN model can be fitted.
- Seasonality was present in the data, therefore **SARIMA** model gave better results.

Model Summary of SARIMA :

SARIMAX Results

Dep. Variable:	Passenger Count	**No. Observations:**	156
Model:	SARIMAX(2, 1, 1)x(0, 1, [], 12)	**Log Likelihood**	-1842.014
Date:	Tue, 05 Dec 2023	**AIC**	3692.028
Time:	03:06:18	**BIC**	3703.879
Sample:	07-01-2005 - 06-01-2018	**HQIC**	3696.844

Covariance Type: opg

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9479	0.025	38.102	0.000	0.899	0.997
ar.L2	0.0096	0.024	0.393	0.694	-0.038	0.057
ma.L1	-0.9998	0.092	-10.893	0.000	-1.180	-0.820
sigma2	8.149e+09	1.13e-11	7.22e+20	0.000	8.15e+09	8.15e+09

Ljung-Box (L1) (Q):	11.95	**Jarque-Bera (JB):**	8.76
Prob(Q):	0.00	**Prob(JB):**	0.01
Heteroskedasticity (H):	1.04	**Skew:**	0.32
Prob(H) (two-sided):	0.89	**Kurtosis:**	4.03

CONCLUSION AND COLAB LINK

The final selected model was SARIMA.
The results of the SARIMA model were compared on different error measurements

Colab Link:
https://colab.research.google.com/drive/1sckOVjJB-LpEHJQegFrNjh10zwpXtEat#scrollTo=-nBwEiF_2440

FUTURE SCOPE

Our optimized SARIMA Model does not take into account the impact of covid 19
Intervention analysis can be done on the future data of this dataset for a comprehensive analysis of TS data