Airline Passenger Demand Forecasting Using ARIMA

Abstract

In this project, a comprehensive time series analysis was performed on monthly airline passenger data collected over several years. The objective was to analyze long-term growth trends, seasonal patterns, and random variations in passenger demand. The study involved visual inspection of the time series, stationarity testing using the Augmented Dickey-Fuller (ADF) test, differencing techniques, autocorrelation and partial autocorrelation analysis (ACF & PACF), and ARIMA model development. The results revealed a strong upward trend, clear yearly seasonality, and non-stationarity in the original series. After applying appropriate differencing, the data became stationary and suitable for ARIMA modeling. This analysis enables accurate forecasting of future passenger demand for effective airline resource planning.

Problem Statement

Airline passenger demand is influenced by various time-dependent factors such as economic growth, seasonal travel trends, holidays, and tourism patterns. Without analyzing the time series structure, forecasting models may generate inaccurate predictions, leading to inefficient aircraft utilization and operational losses.

The goal of this project is to analyze airline passenger data to:

- Identify long-term growth trends
- Detect seasonal patterns
- Measure temporal dependence
- Test stationarity conditions
- Develop reliable ARIMA forecasting models

Table of Contents

Dataset Loading and Preprocessing

The airline passenger dataset was loaded using the Pandas library. The Month column was converted into datetime format and set as the time index to enable time series operations. The frequency of the dataset was set to monthly to ensure consistency.

Key preprocessing steps:

- Converted Month column to datetime format
- Set Month as time index
- Assigned monthly frequency
- Checked for missing values

These steps ensured the dataset was properly structured for ARIMA modeling.

Time Series Visualization

A time series plot was generated to observe passenger demand over time.

Observations:

- A steady increase in passenger numbers over years
- Clear seasonal peaks during specific months
- Repetitive yearly patterns
- Minor short-term fluctuations

The visualization confirmed the presence of both trend and seasonality in the data.

Stationarity Testing and Differencing

To verify whether the time series was stationary, the Augmented Dickey-Fuller (ADF) test was applied.

Initial Results:

- ADF Statistic ≈ −1.35
- p-value ≈ 0.60

Since the p-value was greater than 0.05, the null hypothesis of non-stationarity could not be rejected.

To make the series stationary, first-order differencing was applied.

After Differencing:

- Trend was removed
- Mean and variance became stable
- Series became suitable for modeling

This step was essential for ARIMA implementation.

Autocorrelation Function (ACF) Analysis

The ACF plot was used to examine the correlation between passenger values and their past observations.

Findings:

- High correlation at early lags
- Gradual decay pattern
- Periodic spikes at seasonal intervals

These patterns indicated strong temporal dependence and seasonal behavior in the dataset.

Partial Autocorrelation Function (PACF) Analysis

PACF analysis was performed to measure direct relationships between current and lagged values.

Observations:

- Significant spikes at initial lags
- Rapid decline after certain lags
- Limited influence beyond early lags

These results helped in selecting appropriate autoregressive (AR) parameters for the ARIMA model.

ARIMA Model Development

Based on ACF and PACF analysis, an ARIMA(2,1,2) model was selected for forecasting.

Model Components:

- p = 2 (Auto-Regressive terms)
- d = 1 (Differencing order)
- q = 2 (Moving Average terms)

The model was trained using 80% of the dataset, while the remaining 20% was reserved for testing.

Model Evaluation and Forecasting

Model performance was evaluated using statistical accuracy measures.

Evaluation Metrics:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)

Results indicated satisfactory predictive accuracy.

The trained model was used to forecast passenger demand for the next 24 months.

Forecast Analysis:

• Continued upward trend
• Recurring seasonal patterns
• Stable long-term growth

These forecasts assist airlines in capacity planning and scheduling.

Interpretation of Results

The analysis confirmed that airline passenger demand exhibits:

• Strong upward growth trend
• Clear yearly seasonality
• High temporal dependence
• Initial non-stationarity

After differencing, the dataset satisfied stationarity conditions, enabling effective ARIMA modeling. The forecasting results were consistent with historical patterns, validating the reliability of the model.

Summary

This project successfully analyzed airline passenger data using time series techniques and ARIMA modeling. The dataset was found to be non-stationary due to trend and seasonality. Through differencing and correlation analysis, a suitable ARIMA model was developed and evaluated. The forecasting results provide valuable insights for airline management in optimizing resources and improving operational efficiency.

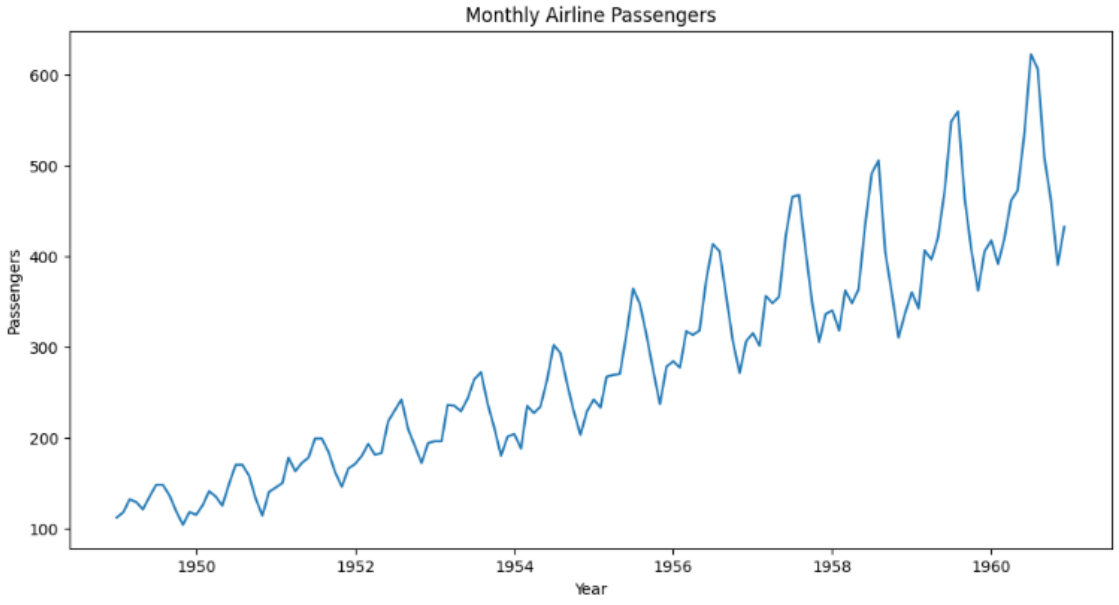
Implementation:

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

[18]:
```
data = pd.read_csv(r"E:\NLP_DATASET\airline_passengers_dataset\airline_passengers.csv")
print(data.columns)
```

```
Index(['Month', 'Passengers'], dtype='object')
```

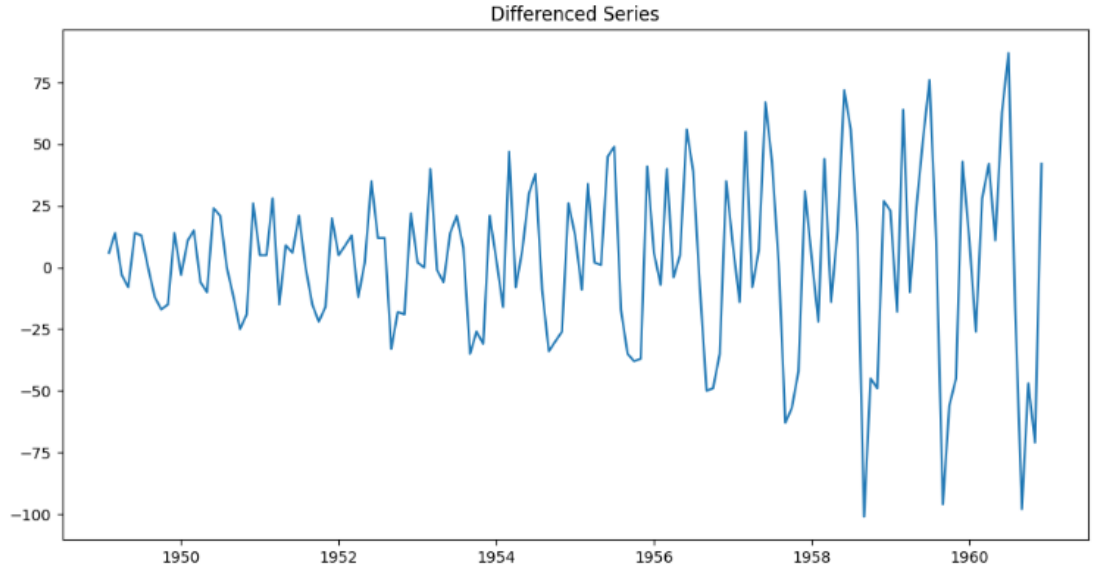
[19]:
```
data["Month"] = pd.to_datetime(data["Month"])

data = data.set_index("Month")

data = data.asfreq("MS")

plt.figure(figsize=(12,6))
plt.plot(data["Passengers"])
plt.title("Monthly Airline Passengers")
plt.xlabel("Year")
plt.ylabel("Passengers")
plt.show()
```



[20]:
```
result = adfuller(data["Passengers"])
print("ADF Statistic:", result[0])
print("p-value:", result[1])
```

```
ADF Statistic: 0.8153688792060482
p-value: 0.991880243437641
```

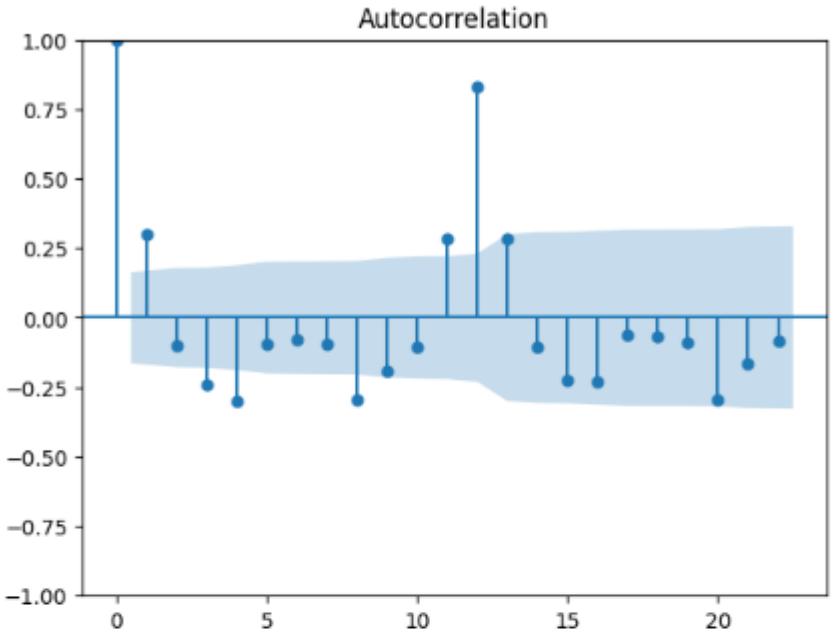
[21]:
```
diff_data = data["Passengers"].diff().dropna()

plt.figure(figsize=(12,6))
plt.plot(diff_data)
plt.title("Differenced Series")
plt.show()
```



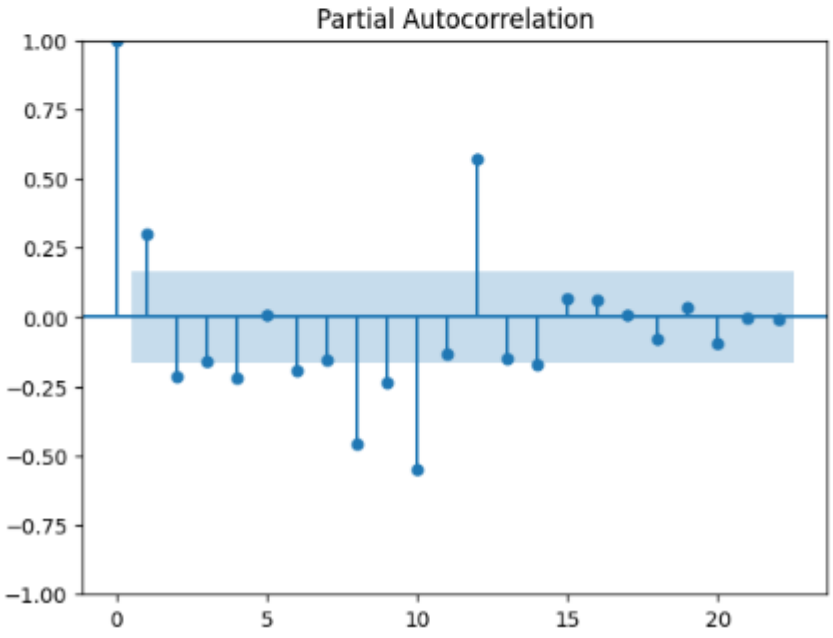
[22]:
```
result_diff = adfuller(diff_data)
print("ADF Statistic After Differencing:", result_diff[0])
print("p-value After Differencing:", result_diff[1])
```

```
ADF Statistic After Differencing: -2.8292668241699994
p-value After Differencing: 0.0542132902838255
```

[23]:
```
plot_acf(diff_data)
plt.show()
```



[24]:
```
plot_pacf(diff_data)
plt.show()
```



[26]:
```
train_size = int(len(data) * 0.8)

train = data.iloc[:train_size]
test = data.iloc[train_size:]

model = ARIMA(train["Passengers"], order=(2,1,2))
model_fit = model.fit()
print(model_fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:             Passengers   No. Observations:                  115
Model:                 ARIMA(2, 1, 2)   Log Likelihood                -523.758
Date:                Fri, 06 Feb 2026   AIC                           1057.516
Time:                        13:40:25   BIC                           1071.197
Sample:                    01-01-1949   HQIC                          1063.069
                         - 07-01-1958
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.3280      0.145      2.268      0.023       0.045       0.611
ar.L2          0.2521      0.165      1.528      0.126      -0.071       0.575
ma.L1         -0.0125      0.109     -0.114      0.909      -0.227       0.202
ma.L2         -0.7544      0.130     -5.812      0.000      -1.009      -0.500
sigma2       568.4920    103.877      5.473      0.000     364.897     772.087
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                 3.39
Prob(Q):                              0.90   Prob(JB):                         0.18
Heteroskedasticity (H):               5.24   Skew:                             0.11
Prob(H) (two-sided):                  0.00   Kurtosis:                         2.19
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

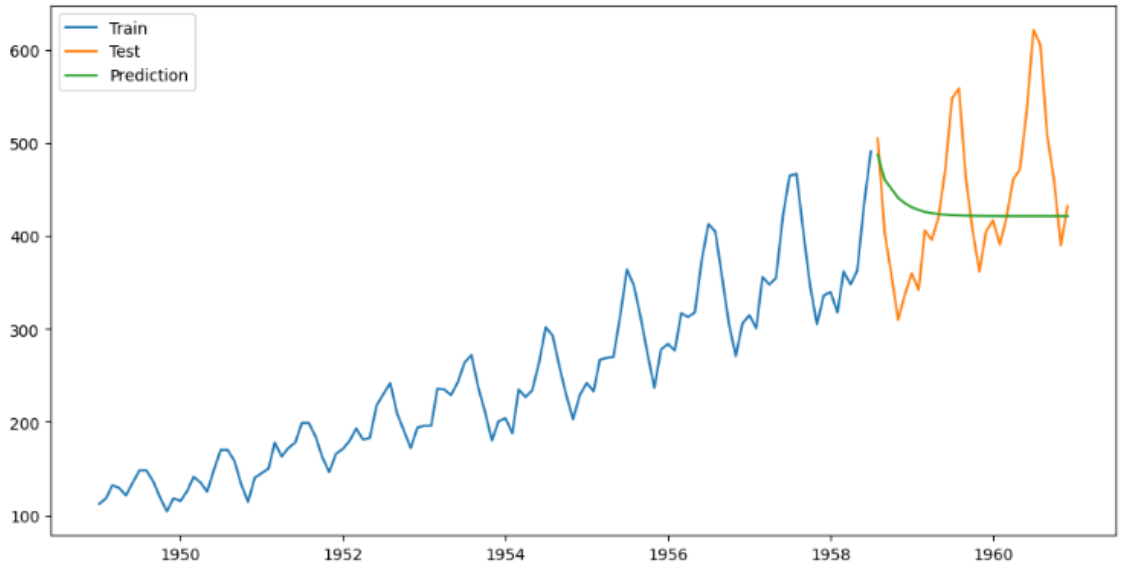
[31]:
```
import warnings
warnings.filterwarnings("ignore")

start = len(train)
end = len(train) + len(test) - 1

pred = model_fit.predict(start=start, end=end, typ="levels")

mae = mean_absolute_error(test["Passengers"], pred)
rmse = np.sqrt(mean_squared_error(test["Passengers"], pred))

print("MAE:", mae)
print("RMSE:", rmse)
```

```
MAE: 63.54531129227126
RMSE: 82.51301128388961
```

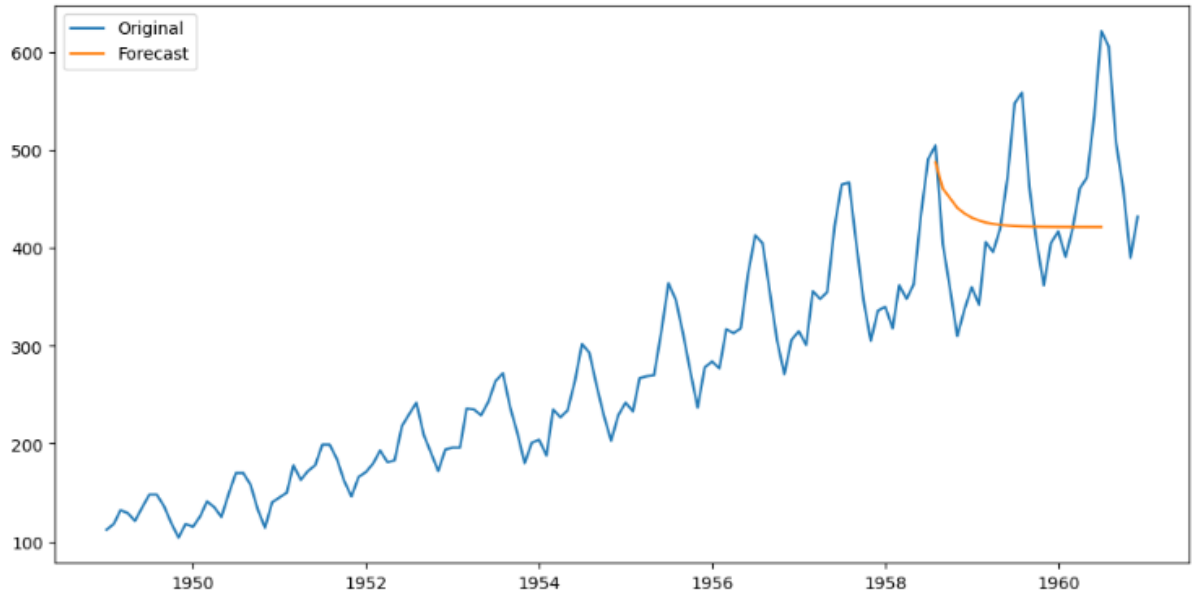
[31]:
```
import warnings
warnings.filterwarnings("ignore")

start = len(train)
end = len(train) + len(test) - 1

pred = model_fit.predict(start=start, end=end, typ="levels")

mae = mean_absolute_error(test["Passengers"], pred)
rmse = np.sqrt(mean_squared_error(test["Passengers"], pred))

print("MAE:", mae)
print("RMSE:", rmse)
```

```
MAE: 63.54531129227126
RMSE: 82.51301128388961
```

[32]:
```
plt.figure(figsize=(12,6))
plt.plot(train["Passengers"], label="Train")
plt.plot(test["Passengers"], label="Test")
plt.plot(pred, label="Prediction")
plt.legend()
plt.show()
```



[29]:
```
future = model_fit.forecast(steps=24)

plt.figure(figsize=(12,6))
plt.plot(data["Passengers"], label="Original")
plt.plot(future, label="Forecast")
plt.legend()
plt.show()
```



[30]:
```
print("Next 24 Months Forecast:")
print(future)
```

```
Next 24 Months Forecast:
1958-08-01    487.825560
1958-09-01    460.796800
1958-10-01    451.130922
1958-11-01    441.145635
1958-12-01    435.433346
1959-01-01    431.042077
1959-02-01    428.161471
1959-03-01    426.109440
1959-04-01    424.710071
1959-05-01    423.733689
1959-06-01    423.060606
1959-07-01    422.593654
1959-08-01    422.270786
1959-09-01    422.047151
1959-10-01    421.892392
1959-11-01    421.785245
1959-12-01    421.711080
1960-01-01    421.659739
1960-02-01    421.624199
1960-03-01    421.599597
1960-04-01    421.582567
1960-05-01    421.570778
1960-06-01    421.562618
1960-07-01    421.556968
Freq: MS, Name: predicted_mean, dtype: float64
```