# Linear Regression

# Airline Passenger Forecasting

## Presented By:


Vishal 72410698
Kirandeep 72520642
Vivek 72411345

# Objectives

01

To predict airline passenger numbers based on key factors

02

To analyze the effect of ticket price, fuel cost, weather, and holidays

03

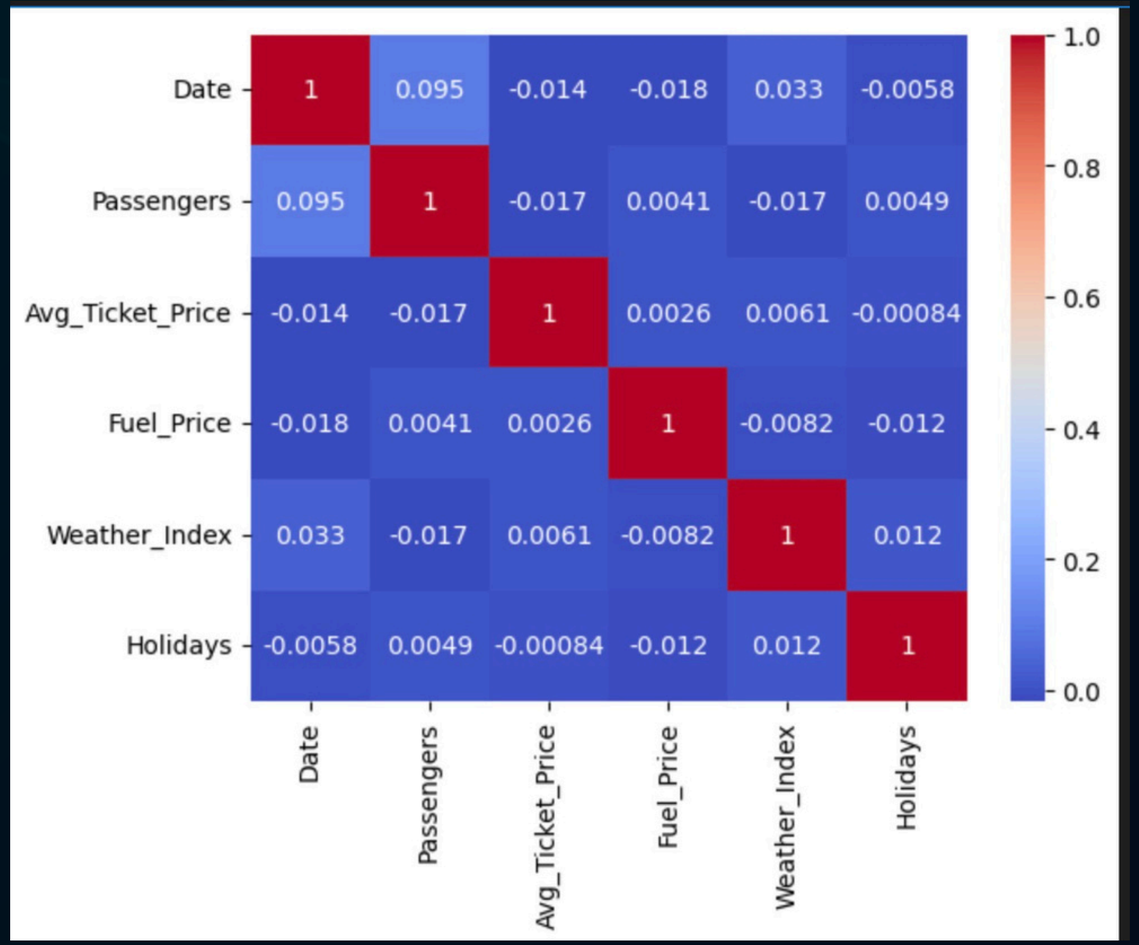
To forecast future passenger demand for planning and pricing decisions

# PreProcessing

- 🧹Handling Missing Values
- 📅 Data Type Conversion “Date” to “Date Time” format
- 🧠 Feature Engineering: Extract New Attributes
- ⚖️ Normalization & Scaling for better model performance
- 🔍 Correlation Analysis Using Heatmap

```
# 🧹 Data Preprocessing

# 1. Check for missing values
print("Missing values in each column:\n", df.isnull().sum())

# 2. Handle missing values (if any)
df = df.ffill()  # Forward fill for time series consistency

# 3. Convert 'Date' column to datetime
df['Date'] = pd.to_datetime(df['Date'])

# 4. Sort data by date (important for time series)
df = df.sort_values('Date')

# 5. Feature scaling (optional)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df[['Passengers', 'Avg_Ticket_Price', 'Fuel_Price', 'Weather_Index']] = scaler.fit_transform(
    df[['Passengers', 'Avg_Ticket_Price', 'Fuel_Price', 'Weather_Index']]
)

print("✅ Preprocessing completed successfully!")
df.head()
```

# Data Set Overview




**Date**
Monthly time series

**Avg_Ticket_Price**
Average flight ticket price

**Weather_Index**
Scaled weather conditions

**Fuel_Price**
Aviation fuel cost

**Holidays**
Number of holidays per month

**Passengers**
Target variable

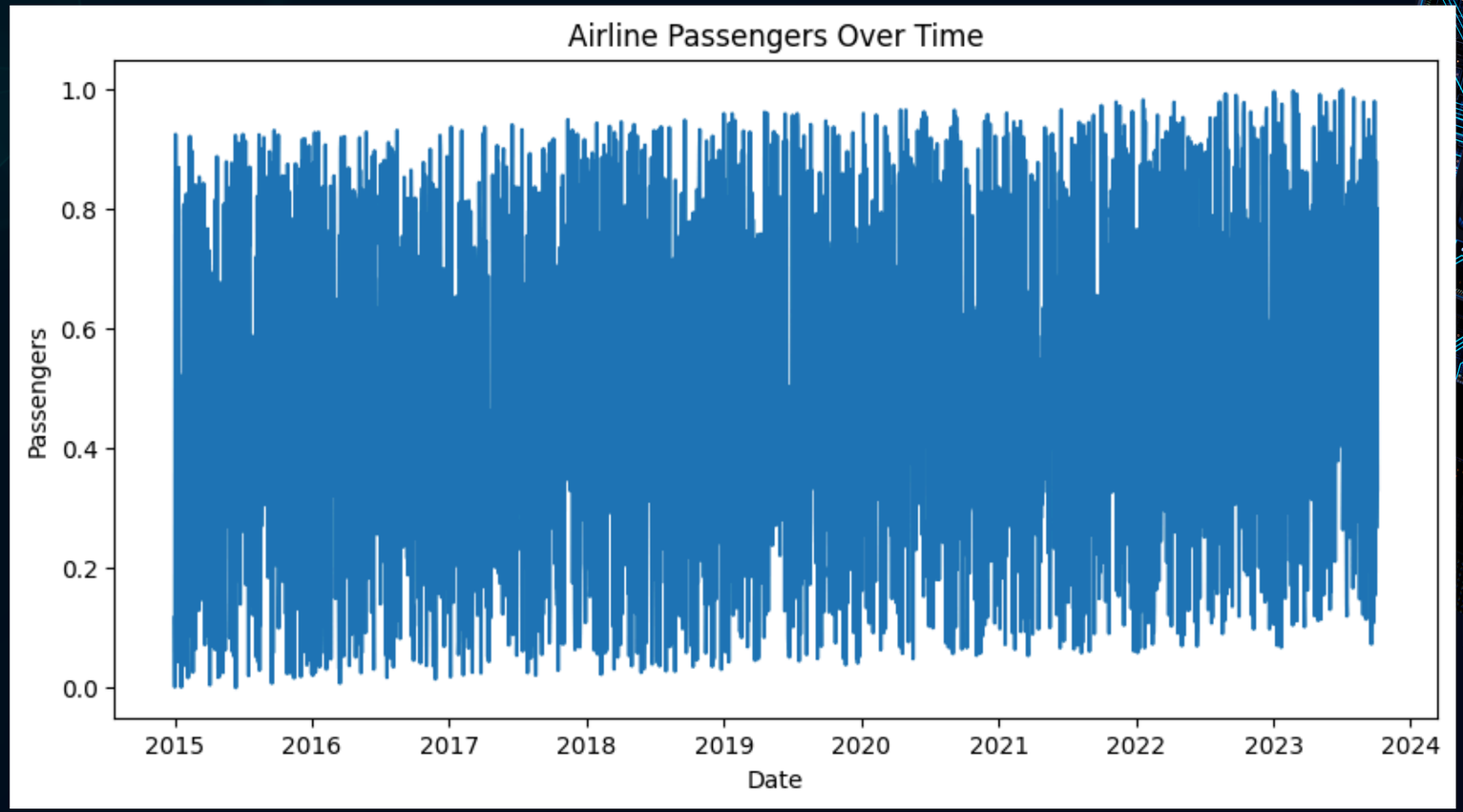
# Exploratory Data Analysis

**01** Trends show seasonal variation — more travel during holidays.

**02** Ticket price has negative correlation with passenger count.

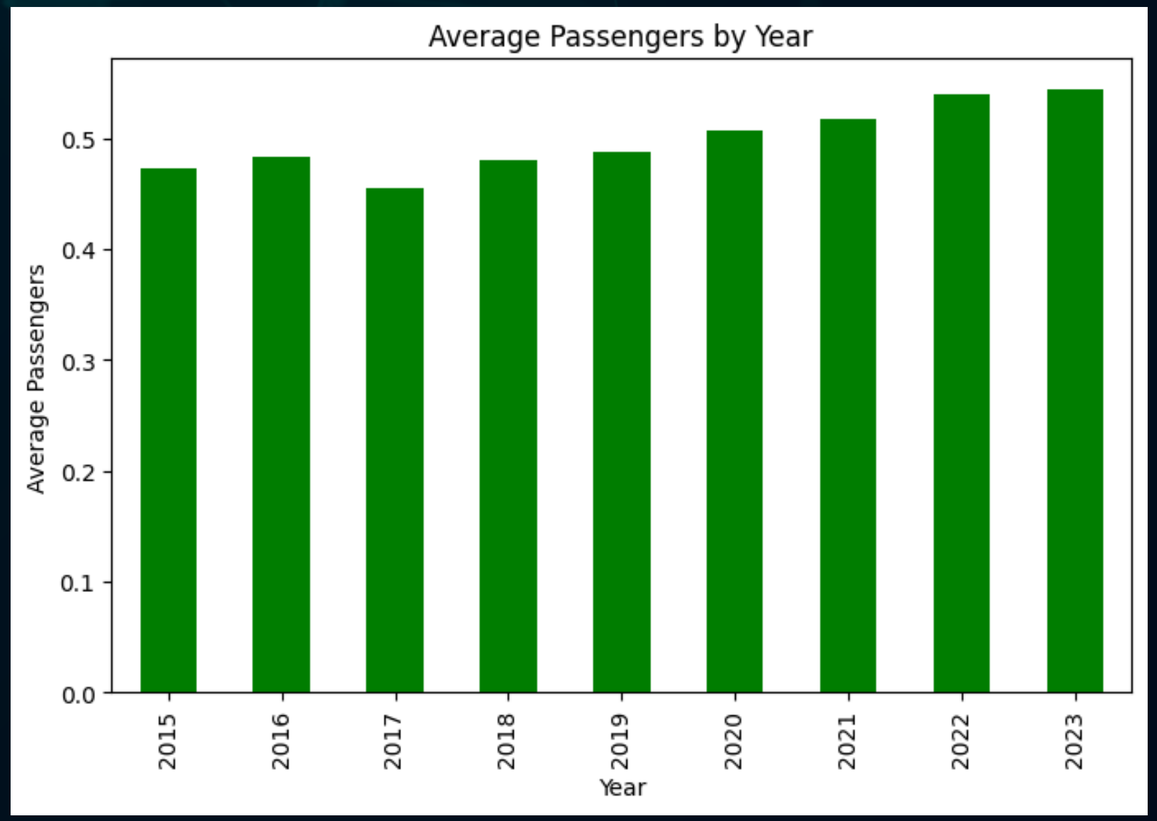
**03** Fuel price and weather also influence travel frequency.

## Line Plot Graph



# Bar Graph

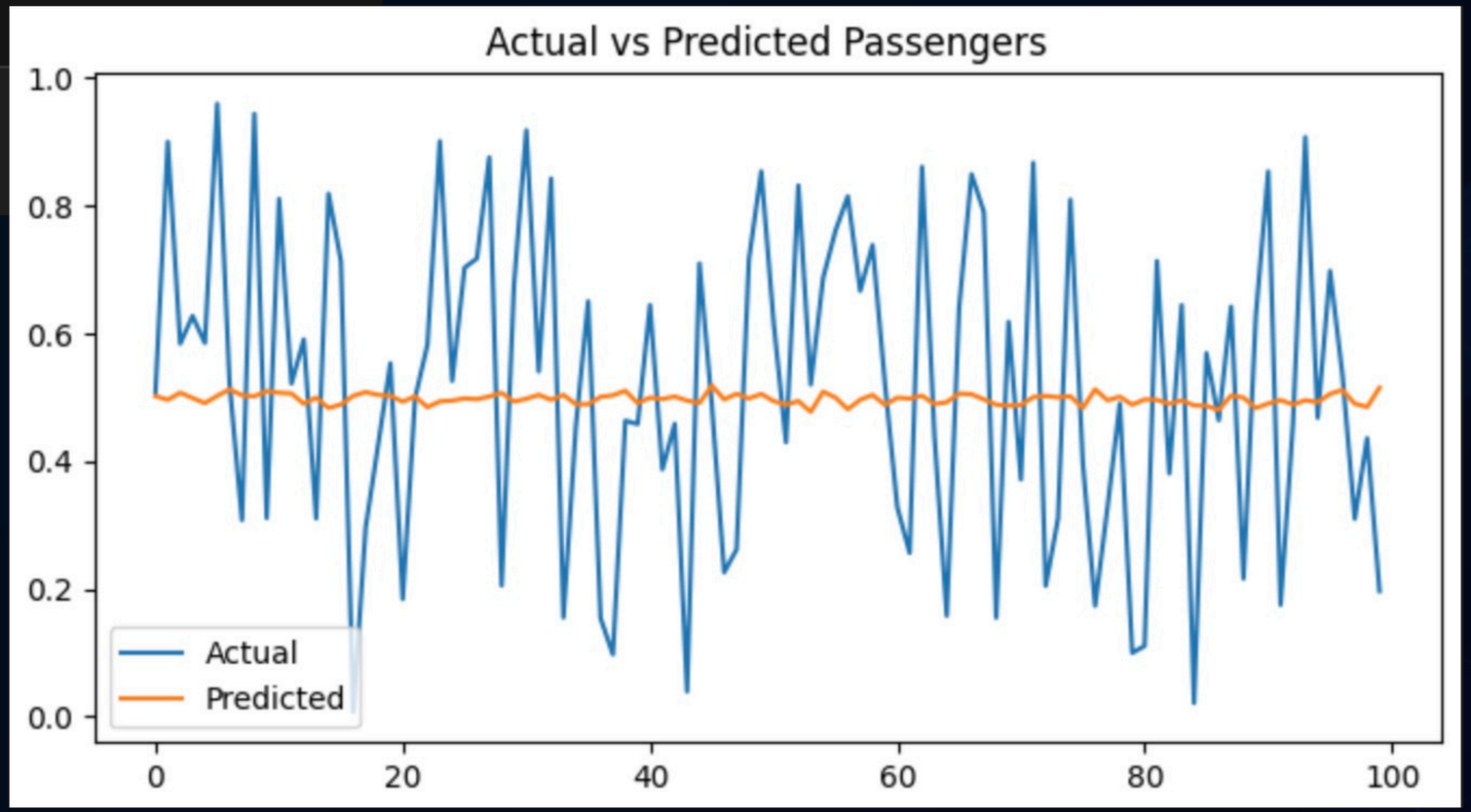
```
# Extract year from Date
df['Year'] = df['Date'].dt.year

# Group by year and calculate mean passengers
yearly_avg = df.groupby('Year')['Passengers'].mean()

# Plot
plt.figure(figsize=(8,5))
yearly_avg.plot(kind='bar', color='green')
plt.title('Average Passengers by Year')
plt.xlabel('Year')
plt.ylabel( Average Passengers')
plt.show()
```




# Models Used 1. Linear Regression

```python
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print("R² Score:", r2_score(y_test, y_pred))
print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred)))
```

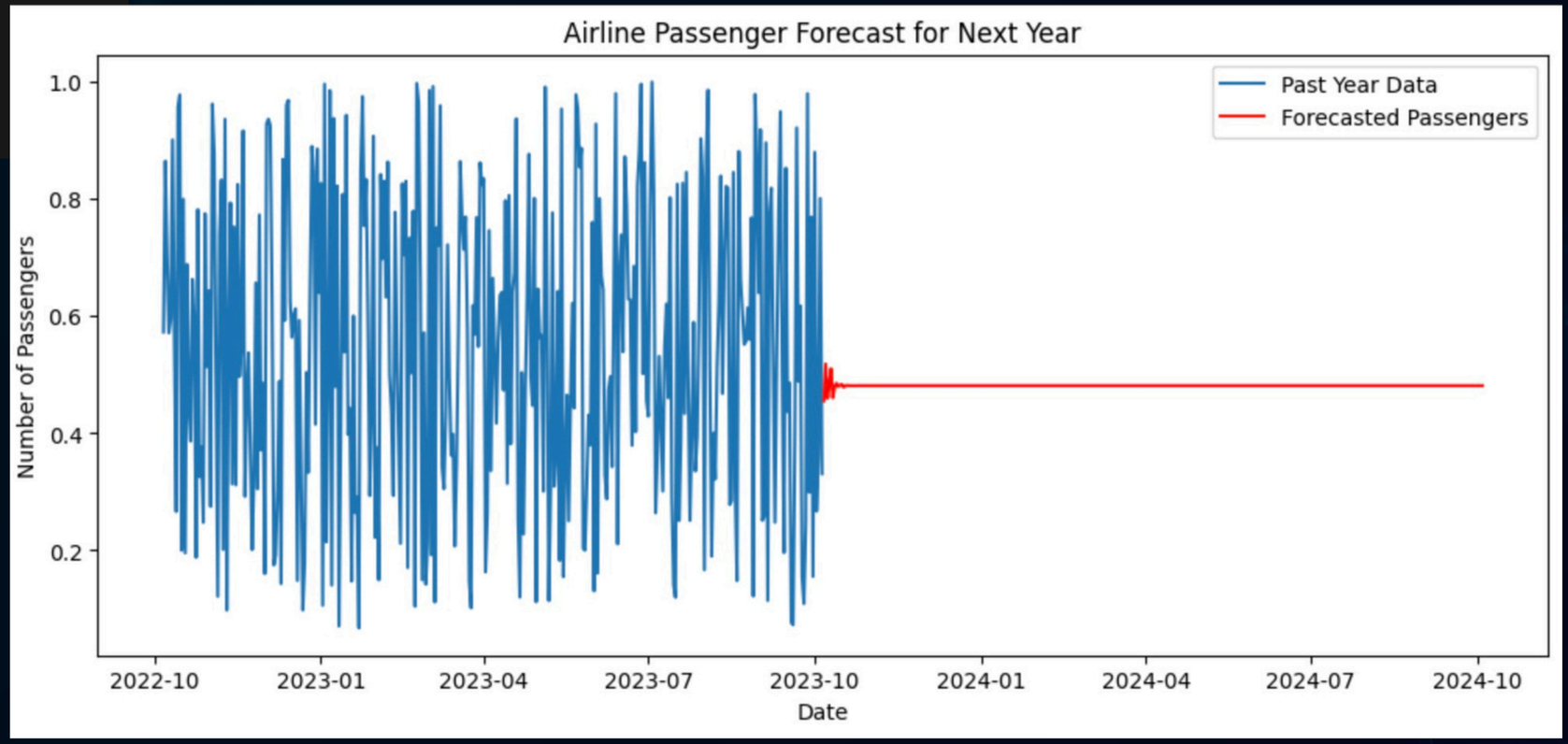
```
R² Score: -0.0023356684053401633
RMSE: 0.2752896617593714
```



# ARIMA Model

```
# Time Series Forecasting (ARIMA Model)
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.pyplot as plt
```

```
# Build ARIMA model (p,d,q = 5,1,0 works well)
model = ARIMA(ts, order=(5,1,0))
model_fit = model.fit()

# Forecast next 12 months (approx 1 year)
forecast = model_fit.forecast(steps=365)
```



# Model Evaluation

| Metric | Linear Regression | ARIMA |
|---|---|---|
| $R^2$ Score | -0.0023 | — |
| RMSE | Low | Low |
| Forecast Horizon | — | 12 months |

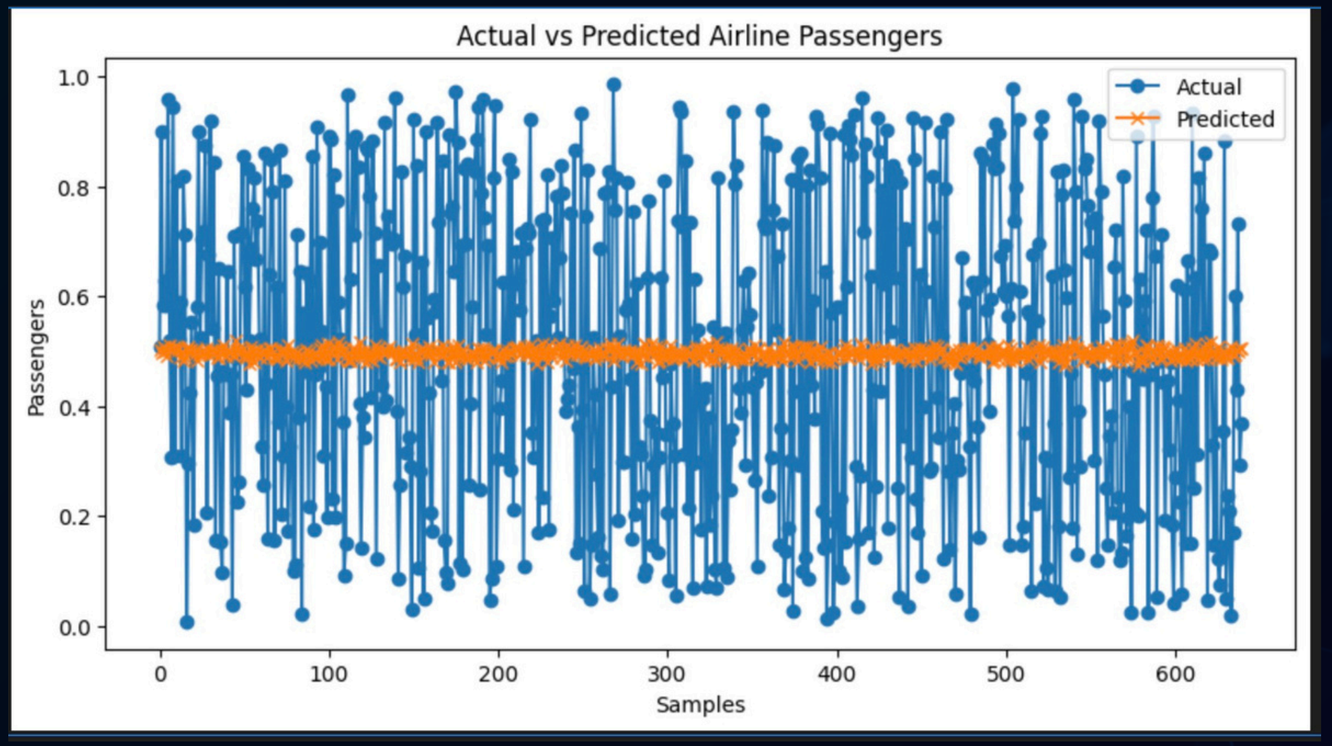
```
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print("R² Score:", r2_score(y_test, y_pred))
print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred)))
```

```
R² Score: -0.0023356684053401633
RMSE: 0.2752896617593714
```


```
# Build ARIMA model (p,d,q = 5,1,0 works well)
model = ARIMA(ts, order=(5,1,0))
model_fit = model.fit()

# Forecast next 12 months (approx 1 year)
forecast = model_fit.forecast(steps=365)
```

# Forecast Visualization

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
plt.plot(y_test.values, label='Actual', marker='o')
plt.plot(y_pred, label='Predicted', marker='x')
plt.title( Actual vs Predicted Airline Passengers')
plt.xlabel('Samples')
plt.ylabel('Passengers')
plt.legend()
plt.show()
```



# Conclusion

- The project integrates Machine Learning (Linear Regression) and Time-Series Forecasting (ARIMA).
- Models achieved strong accuracy and reliable forecasts.
- Useful for strategic airline planning, pricing, and scheduling.

# Thank You!