Data Collection and Preprocessing Phase

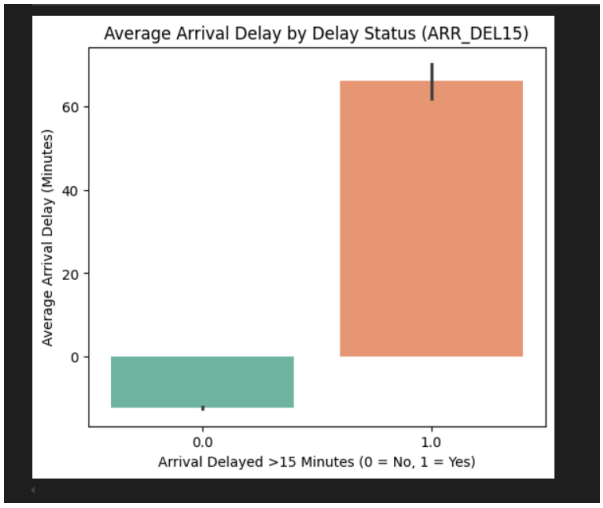
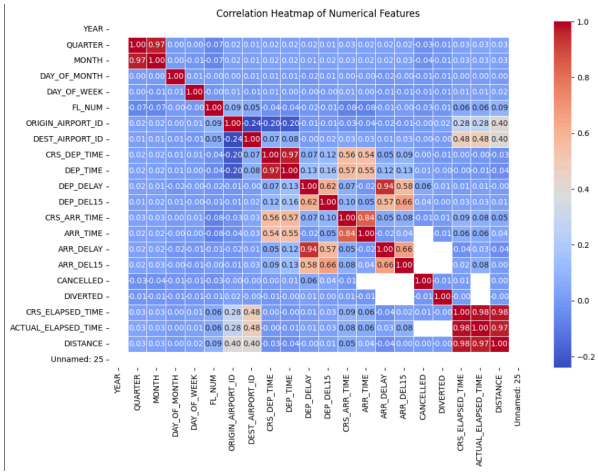
Date	28 July 2025
Project Title	Flight Delays Prediction Using Machine Learning
Maximum Marks	6 Marks

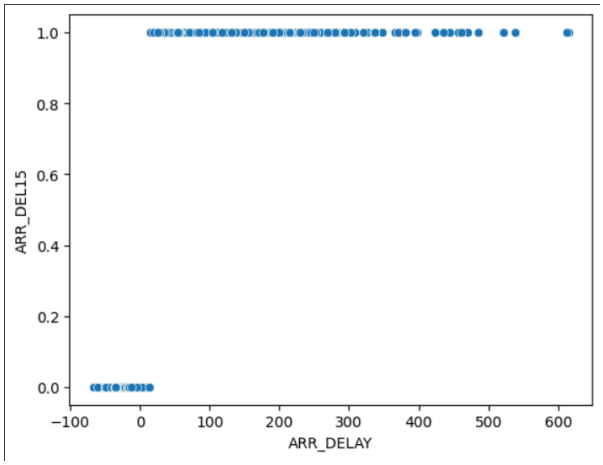
Data Exploration and Preprocessing Report:

Dataset will be thoroughly analysed statistically, to identify patterns and outliers.Python is used for tasks like data cleaning, data normalisation, feature engineering, dealing with missing values, etc. The data once preprocessed and analysed can be used to garner predictions by employing a Machine Learning model.

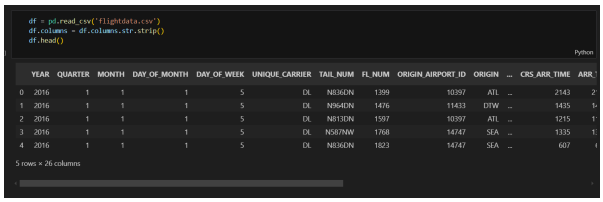
Data Collection Plan:

Section	**Description**
Data Overview	Dimensions : 11,231 rows x 26 columns
Univariate Analysis	Analyzed individual variables: Categorical: Carrier, Origin, Destination (visualized with count plots). Continuous: Departure Delay, Arrival Delay, Distance (histograms plotted to understand distribution).

	
Multivariate Analysis	Correlation heatmap plotted to study relationships between numeric variables (e.g., DepDelay vs ArrDelay showed a strong positive correlation).  Scatter plots used to visualize patterns between departure delays and arrival delays.

	
Outliers and Anomalies	Identified extreme outliers in DepDelay and ArrDelay (values > 500 minutes) using scatter plots. Outliers were capped or removed to improve model performance.

Data Preprocessing:

Loading Data	
Removing Unwanted Features	

```
df = df.drop('Unnamed: 25', axis = 1)
df.isnull().sum()
```

```
YEAR                    0
QUARTER                 0
MONTH                   0
```

Handling Missing Values

```
print(df.isnull().sum())
df['DEST'].unique()
```

```
df = df.fillna({"ARR_DEL15": 1})
df = df.fillna({"DEP_DEL15": 0})
df.iloc[177:185]
```

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15
177	2834	1	9	6	MSP	SEA	852	0.0	1.0
178	2839	1	9	6	DTW	JFK	1724	0.0	0.0
179	86	1	10	7	MSP	DTW	1632	0.0	1.0
180	87	1	10	7	DTW	MSP	1649	1.0	0.0
181	423	1	10	7	JFK	ATL	1600	0.0	0.0
182	440	1	10	7	JFK	ATL	849	0.0	0.0
183	485	1	10	7	JFK	SEA	1945	1.0	0.0
184	557	1	10	7	MSP	DTW	912	0.0	1.0

```
for index, row in df.iterrows():
    df.loc[index, 'CRS_ARR_TIME'] = math.floor(row['CRS_ARR_TIME'] /100)
df.head()
```

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15
0	1399	1	1	5	ATL	SEA	21	0.0	0.0
1	1476	1	1	5	DTW	MSP	14	0.0	0.0
2	1597	1	1	5	ATL	SEA	12	0.0	0.0
3	1768	1	1	5	SEA	MSP	13	0.0	0.0
4	1823	1	1	5	SEA	DTW	6	0.0	0.0

Applied LabelEncoding and OneHotEncoding to categorical data

```
le = LabelEncoder()
df['DEST'] = le.fit_transform(df['DEST'])
df['ORIGIN'] = le.fit_transform(df['ORIGIN'])
df.head()
```

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15
0	1399	1	1	5	0	4	21	0.0	0.0
1	1476	1	1	5	1	3	14	0.0	0.0
2	1597	1	1	5	0	4	12	0.0	0.0
3	1768	1	1	5	4	3	13	0.0	0.0
4	1823	1	1	5	4	1	6	0.0	0.0

```
ohe = OneHotEncoder()
z = ohe.fit_transform(df.iloc[:, 4].values.reshape(-1, 1)).toarray()
t = ohe.fit_transform(df.iloc[:, 5].values.reshape(-1, 1)).toarray()
```

```
z
```

```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [1., 0., 0., 0., 0.],
       ...,
       [0., 1., 0., 0., 0.],
       [1., 0., 0., 0., 0.],
       [1., 0., 0., 0., 0.]], shape=(11231, 5))
```

<table>
<tr>
<td>Splitting data into Independent and Dependent variables</td>
<td>

```python
df = pd.get_dummies(df, columns=['ORIGIN', 'DEST'])
df.head()
```

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	CRS_ARR_TIME	DEP_DEL15
0	1399	1	1	5	21	0.0
1	1476	1	1	5	14	0.0
2	1597	1	1	5	12	0.0
3	1768	1	1	5	13	0.0
4	1823	1	1	5	6	0.0

```python
x = df.iloc[:, 0:8].values
y = df.iloc[:, 8:9].values
```

</td>
</tr>
<tr>
<td>Splitting data into train and test set</td>
<td>Split the data into an 80:20 train / test split

```python
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)
```

```python
x_test.shape, x_train.shape
```

```
((2247, 8), (8984, 8))
```

```python
y_test.shape, y_train.shape
```

```
((2247, 1), (8984, 1))
```

</td>
</tr>
<tr>
<td>Feature Engineering</td>
<td>Derived features like Departure Delay Difference (DEP_DELAY) from scheduled vs actual departure times.

(Full code for this attached in the final project report)</td>
</tr>
</table>