Flight price data set

Introduction:

This dataset contains detailed records of flights operating in and out of various airports in Bangladesh. It captures multiple dimensions of flight information, ranging from logistics and airline details to pricing structures. The data could be used to explore and model airfare trends, understand the impact of seasonality and booking channels on prices, and predict total fare costs.

Key Features: .Airline & Aircraft Details: Includes the airline name and aircraft type (e.g., Airbus A320, Boeing 787)

.Route Information: Source and destination codes along with airport names.

.Schedule: Departure and arrival date & time.

.Duration: Flight duration in hours.

.Stops: Whether the flight is direct or has stopovers.

.Class: Ticket class (Economy, Business, First Class).

.Booking Source: Where the ticket was booked (e.g., Online Website, Travel Agency).

.Fare Information:

.Base Fare (BDT)

.Tax & Surcharge (BDT)

.Total Fare (BDT)

.Seasonality: Indicates travel season (e.g., Regular, Winter Holidays).

.Days Before Departure: Time gap between booking and flight date.

Model Inplementation:

In this project, a machine learning regression model is used to predict the Total Fare (BDT) of flights in Bangladesh. The dataset includes features like airline, source and destination, duration, class, booking source, taxes, and days before departure. The data is cleaned and preprocessed by handling missing values and encoding categorical variables. Then, the features and target variable are separated, and a regression model (like Random Forest) is trained to learn how different factors influence the fare.

Model Evaluation

The model's performance is evaluated using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). These help measure how accurate the fare predictions are and how well the model captures the relationship between input features and flight prices.

Visualization and Prediction

Visualizations like feature importance charts help understand which factors most affect flight fares. Once trained, the model can predict the fare of a flight based on user inputs, making it useful for analyzing trends and estimating future flight costs.

[65]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

[67]:
```
df = pd.read_csv(r"C:\Users\athul\Downloads\Flight_Price_Dataset_of_Bangladesh.
↪csv")
df
```

[67]:
```
                      Airline  Source  \
0           Malaysian Airlines     CXB
1                Cathay Pacific     BZL
2              British Airways     ZYL
3            Singapore Airlines     RJH
4               British Airways     SPD
...                        ...     ...
56995           Kuwait Airways     JSR
56996           Kuwait Airways     CGP
56997  Biman Bangladesh Airlines     CXB
56998          British Airways     SPD
56999                Air India     DAC

                                         Source Name Destination  \
0                                 Cox's Bazar Airport         CCU
1                                     Barisal Airport         CGP
2                    Osmani International Airport, Sylhet         KUL
3                        Shah Makhdum Airport, Rajshahi         DAC
4                                     Saidpur Airport         YYZ
...                                              ...         ...
56995                                 Jessore Airport         CCU
56996  Shah Amanat International Airport, Chittagong         CCU
56997                             Cox's Bazar Airport         JSR
56998                                 Saidpur Airport         YYZ
56999  Hazrat Shahjalal International Airport, Dhaka         RJH

                                       Destination Name \
0      Netaji Subhas Chandra Bose International Airpo…
1          Shah Amanat International Airport, Chittagong
```

2	Kuala Lumpur International Airport
3	Hazrat Shahjalal International Airport, Dhaka
4	Toronto Pearson International Airport
...	...
56995	Netaji Subhas Chandra Bose International Airpo...
56996	Netaji Subhas Chandra Bose International Airpo...
56997	Jessore Airport
56998	Toronto Pearson International Airport
56999	Shah Makhdum Airport, Rajshahi

	Departure Date & Time	Arrival Date & Time	Duration (hrs)	Stopovers \
0	2025-11-17 06:25:00	2025-11-17 07:38:10	1.219526	Direct
1	2025-03-16 00:17:00	2025-03-16 00:53:31	0.608638	Direct
2	2025-12-13 12:03:00	2025-12-13 14:44:22	2.689651	1 Stop
3	2025-05-30 03:21:00	2025-05-30 04:02:09	0.686054	Direct
4	2025-04-25 09:14:00	2025-04-25 23:17:20	14.055609	1 Stop
...	...	...	...	...
56995	2025-08-11 00:10:00	2025-08-11 00:40:00	0.500000	Direct
56996	2025-09-19 23:53:00	2025-09-20 01:09:30	1.275145	Direct
56997	2025-11-08 09:23:00	2025-11-08 10:35:59	1.216583	Direct
56998	2025-11-25 10:23:00	2025-11-26 00:20:37	13.960502	1 Stop
56999	2025-07-05 04:12:00	2025-07-05 04:50:55	0.648755	Direct

	Aircraft Type	Class	Booking Source	Base Fare (BDT) \
0	Airbus A320	Economy	Online Website	21131.225021
1	Airbus A320	First Class	Travel Agency	11605.395471
2	Boeing 787	Economy	Travel Agency	39882.499349
3	Airbus A320	Economy	Direct Booking	4435.607340
4	Airbus A350	Business	Direct Booking	59243.806146
...	...	...	...	...
56995	Airbus A320	Business	Online Website	79974.471748
56996	Airbus A320	First Class	Online Website	193471.364277
56997	Airbus A320	Economy	Direct Booking	4375.365554
56998	Airbus A350	Economy	Direct Booking	40903.602688
56999	Airbus A320	Business	Direct Booking	5831.070839

	Tax & Surcharge (BDT)	Total Fare (BDT)	Seasonality \
0	5169.683753	26300.908775	Regular
1	200.000000	11805.395471	Regular
2	11982.374902	51864.874251	Winter Holidays
3	200.000000	4635.607340	Regular
4	14886.570922	74130.377068	Regular
...	...	...	...
56995	13996.170762	93970.642511	Regular
56996	31020.704642	224492.068918	Regular
56997	200.000000	4575.365554	Regular
56998	12135.540403	53039.143091	Regular

```
56999          200.000000     6031.070839        Regular

       Days Before Departure
0                         10
1                         14
2                         83
3                         56
4                         90
...                      ...
56995                     51
56996                     31
56997                     22
56998                     20
56999                      6

[57000 rows x 17 columns]
```

[69]:
```
df.head()
```

[69]:
```
             Airline  Source                            Source Name  \
0  Malaysian Airlines    CXB                      Cox's Bazar Airport
1      Cathay Pacific    BZL                          Barisal Airport
2     British Airways    ZYL  Osmani International Airport, Sylhet
3  Singapore Airlines    RJH           Shah Makhdum Airport, Rajshahi
4     British Airways    SPD                          Saidpur Airport

  Destination                                   Destination Name  \
0         CCU  Netaji Subhas Chandra Bose International Airpo...
1         CGP       Shah Amanat International Airport, Chittagong
2         KUL                      Kuala Lumpur International Airport
3         DAC       Hazrat Shahjalal International Airport, Dhaka
4         YYZ                   Toronto Pearson International Airport

  Departure Date & Time Arrival Date & Time  Duration (hrs) Stopovers  \
0   2025-11-17 06:25:00 2025-11-17 07:38:10        1.219526    Direct
1   2025-03-16 00:17:00 2025-03-16 00:53:31        0.608638    Direct
2   2025-12-13 12:03:00 2025-12-13 14:44:22        2.689651    1 Stop
3   2025-05-30 03:21:00 2025-05-30 04:02:09        0.686054    Direct
4   2025-04-25 09:14:00 2025-04-25 23:17:20       14.055609    1 Stop

  Aircraft Type        Class  Booking Source  Base Fare (BDT)  \
0   Airbus A320      Economy  Online Website     21131.225021
1   Airbus A320  First Class    Travel Agency     11605.395471
2    Boeing 787      Economy    Travel Agency     39882.499349
3   Airbus A320      Economy  Direct Booking      4435.607340
4   Airbus A350     Business  Direct Booking     59243.806146
```

```
   Tax & Surcharge (BDT)  Total Fare (BDT)      Seasonality  \
0            5169.683753      26300.908775          Regular
1             200.000000      11805.395471          Regular
2           11982.374902      51864.874251  Winter Holidays
3             200.000000       4635.607340          Regular
4           14886.570922      74130.377068          Regular

   Days Before Departure
0                     10
1                     14
2                     83
3                     56
4                     90
```

[71]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57000 entries, 0 to 56999
Data columns (total 17 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Airline                57000 non-null  object
 1   Source                 57000 non-null  object
 2   Source Name            57000 non-null  object
 3   Destination            57000 non-null  object
 4   Destination Name       57000 non-null  object
 5   Departure Date & Time  57000 non-null  object
 6   Arrival Date & Time    57000 non-null  object
 7   Duration (hrs)         57000 non-null  float64
 8   Stopovers              57000 non-null  object
 9   Aircraft Type          57000 non-null  object
 10  Class                  57000 non-null  object
 11  Booking Source         57000 non-null  object
 12  Base Fare (BDT)        57000 non-null  float64
 13  Tax & Surcharge (BDT)  57000 non-null  float64
 14  Total Fare (BDT)       57000 non-null  float64
 15  Seasonality            57000 non-null  object
 16  Days Before Departure  57000 non-null  int64
dtypes: float64(4), int64(1), object(12)
memory usage: 7.4+ MB
```

[73]:
```
df.describe()
```

[73]:
```
       Duration (hrs)  Base Fare (BDT)  Tax & Surcharge (BDT)  \
count    57000.000000     57000.000000           57000.000000
mean         3.994955     58899.556573           11448.238494
std          4.094043     68840.614499           12124.344329
```

```
min           0.500000     1600.975688             200.000000
25%           1.003745     8856.316983             200.000000
50%           2.644656    31615.996792            9450.940481
75%           5.490104    85722.930389           17513.046160
max          15.831719   449222.933770           73383.440066

       Total Fare (BDT)  Days Before Departure
count      57000.000000           57000.000000
mean       71030.316199              45.460579
std        81769.199536              26.015657
min         1800.975688               1.000000
25%         9602.699787              23.000000
50%        41307.544990              45.000000
75%       103800.906963              68.000000
max       558987.332444              90.000000
```

[75]:
```
df.isnull().sum()
```

[75]:
```
Airline                   0
Source                    0
Source Name               0
Destination               0
Destination  Name         0
Departure Date & Time     0
Arrival Date & Time       0
Duration (hrs)            0
Stopovers                 0
Aircraft  Type            0
Class                     0
Booking Source            0
Base Fare (BDT)           0
Tax & Surcharge (BDT)     0
Total Fare (BDT)          0
Seasonality               0
Days Before Departure     0
dtype: int64
```

[77]:
```
most_common_airline = df['Airline'].mode()[0]

# Fill missing Airline values with the most common one
df['Airline'] = df['Airline'].fillna(most_common_airline)
```

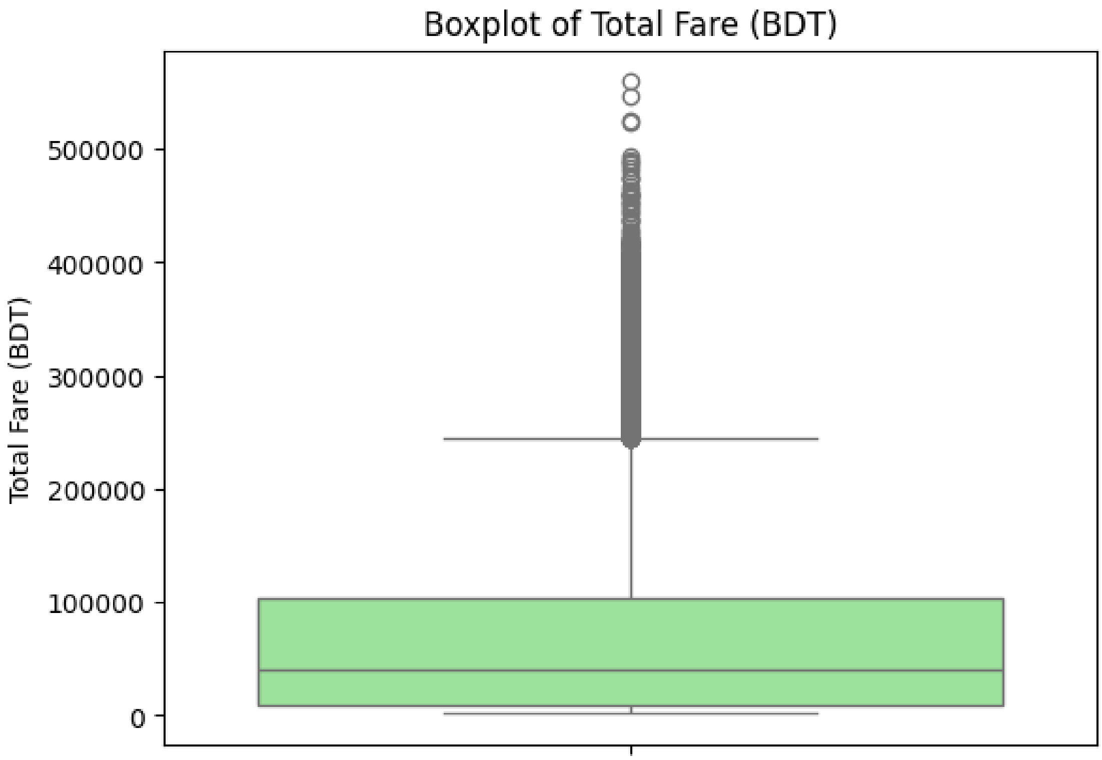
[79]:
```
df['Airline'] = df['Airline'].fillna(df['Airline'].mode()[0])
df['Source'] = df['Source'].fillna(df['Source'].mode()[0])
df['Destination'] = df['Destination'].fillna(df['Destination'].mode()[0])
df['Class'] = df['Class'].fillna(df['Class'].mode()[0])
```

[81]:
```
# Filling missing values in categorical columns with their mode
df['Airline'] = df['Airline'].fillna(df['Airline'].mode()[0])
df['Source'] = df['Source'].fillna(df['Source'].mode()[0])
df['Destination'] = df['Destination'].fillna(df['Destination'].mode()[0])
df['Class'] = df['Class'].fillna(df['Class'].mode()[0])
```

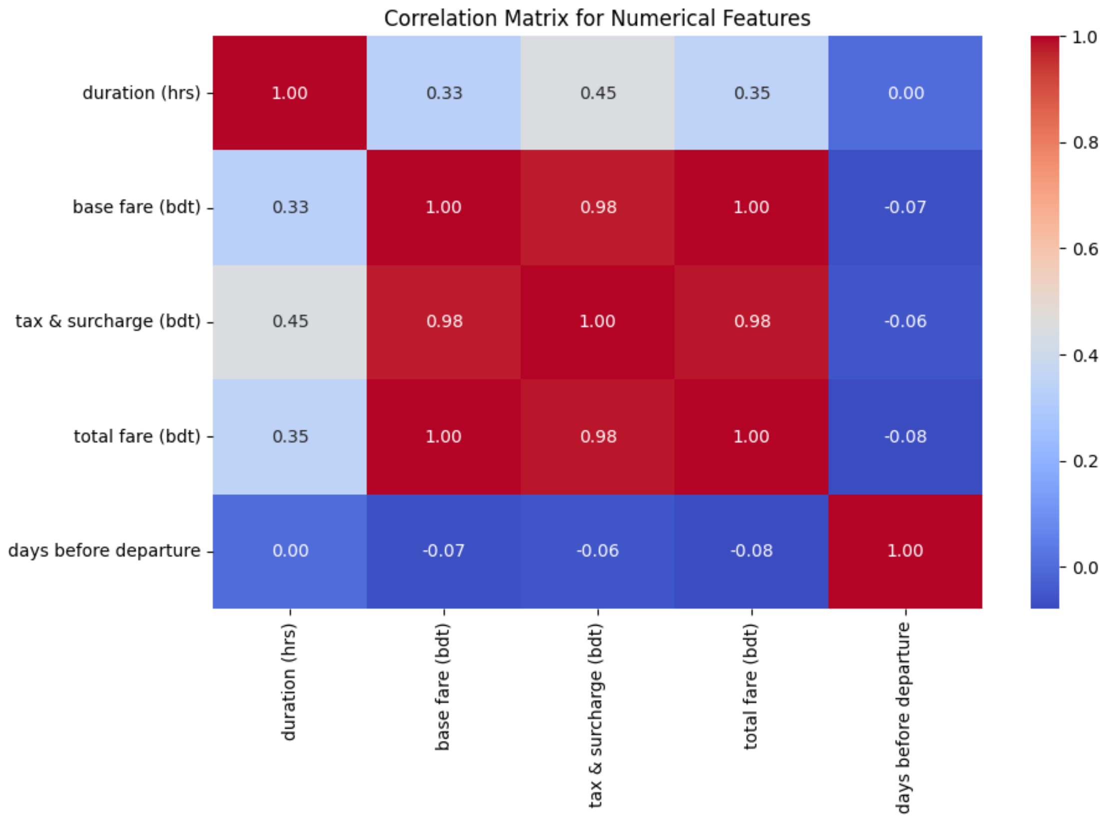
[83]:
```
df.isnull().sum()
```

[83]:
```
Airline                    0
Source                     0
Source Name                0
Destination                0
Destination Name           0
Departure Date & Time      0
Arrival Date & Time        0
Duration (hrs)             0
Stopovers                  0
Aircraft Type              0
Class                      0
Booking Source             0
Base Fare (BDT)            0
Tax & Surcharge (BDT)      0
Total Fare (BDT)           0
Seasonality                0
Days Before Departure      0
dtype: int64
```

[93]:
```
# Then plot the boxplot
sns.boxplot(y=df['total fare (bdt)'], color='lightgreen')
plt.title('Boxplot of Total Fare (BDT)')
plt.ylabel('Total Fare (BDT)')
plt.show()
```

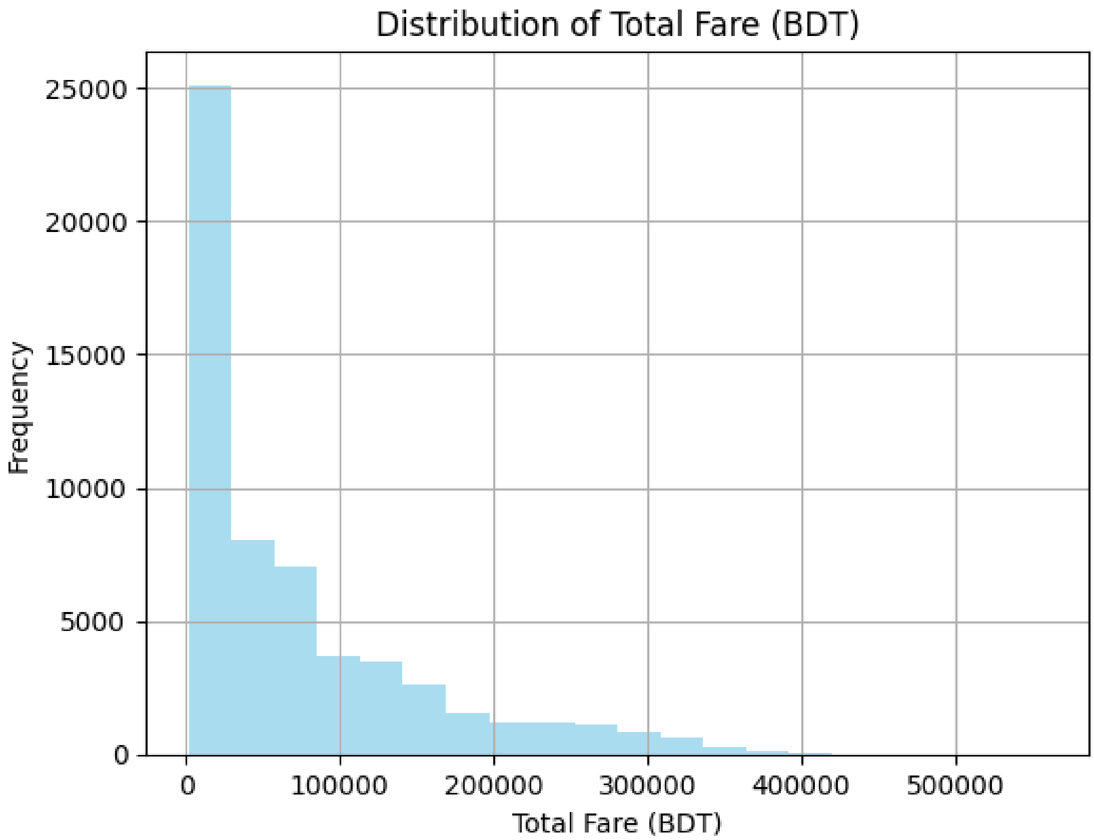


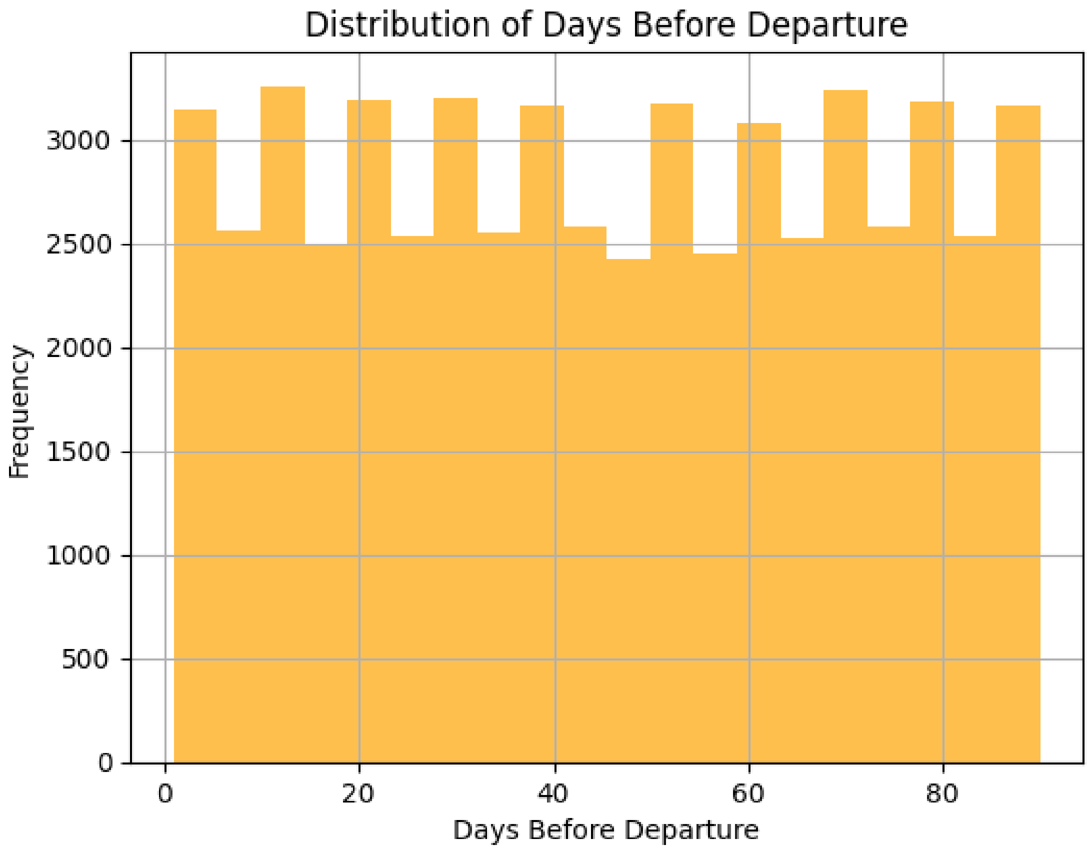
[95]:
```
# Select only numerical columns
numerical_cols = df.select_dtypes(include=['float64', 'int64']).columns

# Calculate the correlation matrix
corr_matrix = df[numerical_cols].corr()

# Plot the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix for Numerical Features')
plt.show()
```



[97]:

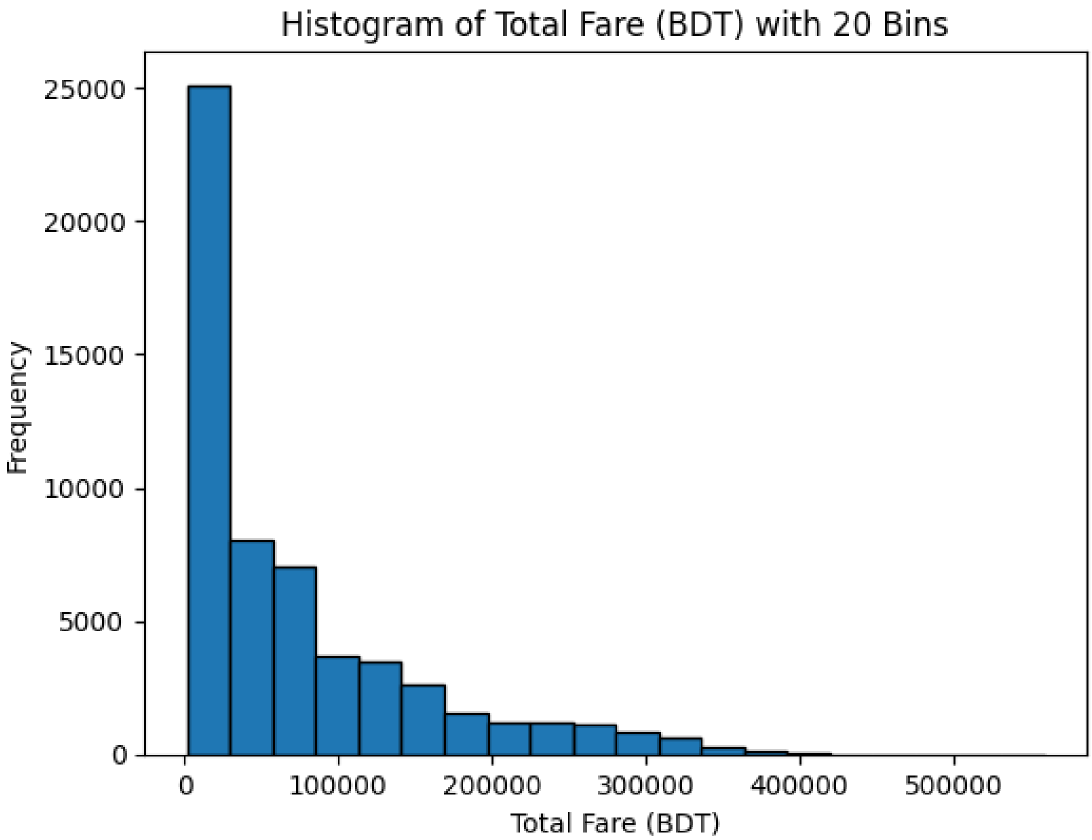
```
# Histogram for Total Fare (as Transaction Amount)
df['total fare (bdt)'].hist(bins=20, color='skyblue', alpha=0.7)
plt.title('Distribution of Total Fare (BDT)')
plt.xlabel('Total Fare (BDT)')
plt.ylabel('Frequency')
plt.show()

# Histogram for Days Before Departure (as Time Since Last Transaction)
df['days before departure'].hist(bins=20, color='orange', alpha=0.7)
plt.title('Distribution of Days Before Departure')
plt.xlabel('Days Before Departure')
plt.ylabel('Frequency')
plt.show()
```


Distribution of Total Fare (BDT)
Frequency
25000
20000
15000
10000
5000
0
0
100000
200000
300000
400000
500000
Total Fare (BDT)



[99]:
```
df.to_csv('cleaned_creditcard_data.csv',  index=False)
```

[103]:
```
# Plot histogram
plt.hist(df['total fare (bdt)'], bins=20, edgecolor='black')
plt.xlabel('Total Fare (BDT)')
plt.ylabel('Frequency')
plt.title('Histogram of Total Fare (BDT) with 20 Bins')
plt.show()
```



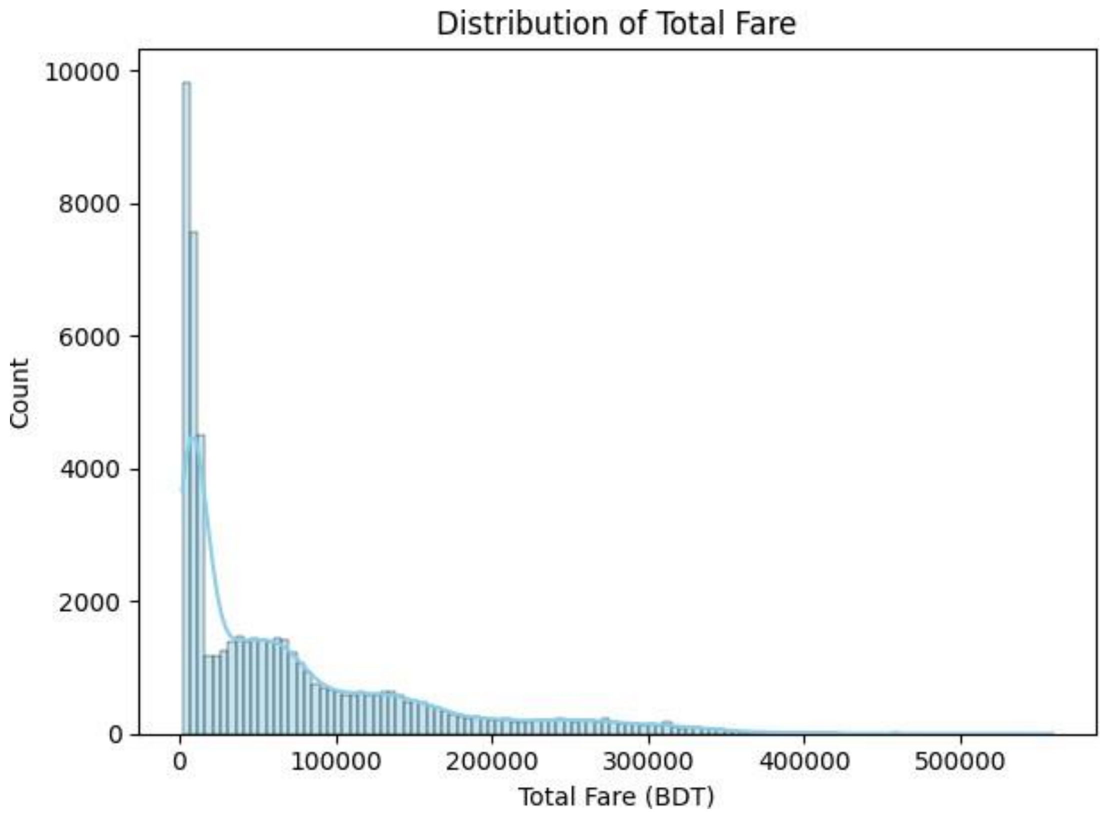
0.1 Univariate Analysis

Univariate Analysis is a type of data visualization where we visualize only a single variable at a time. Univariate Analysis helps us to analyze the distribution of the variable present in the data so that we can perform further analysis. You can find the link to the dataset here

Histogram

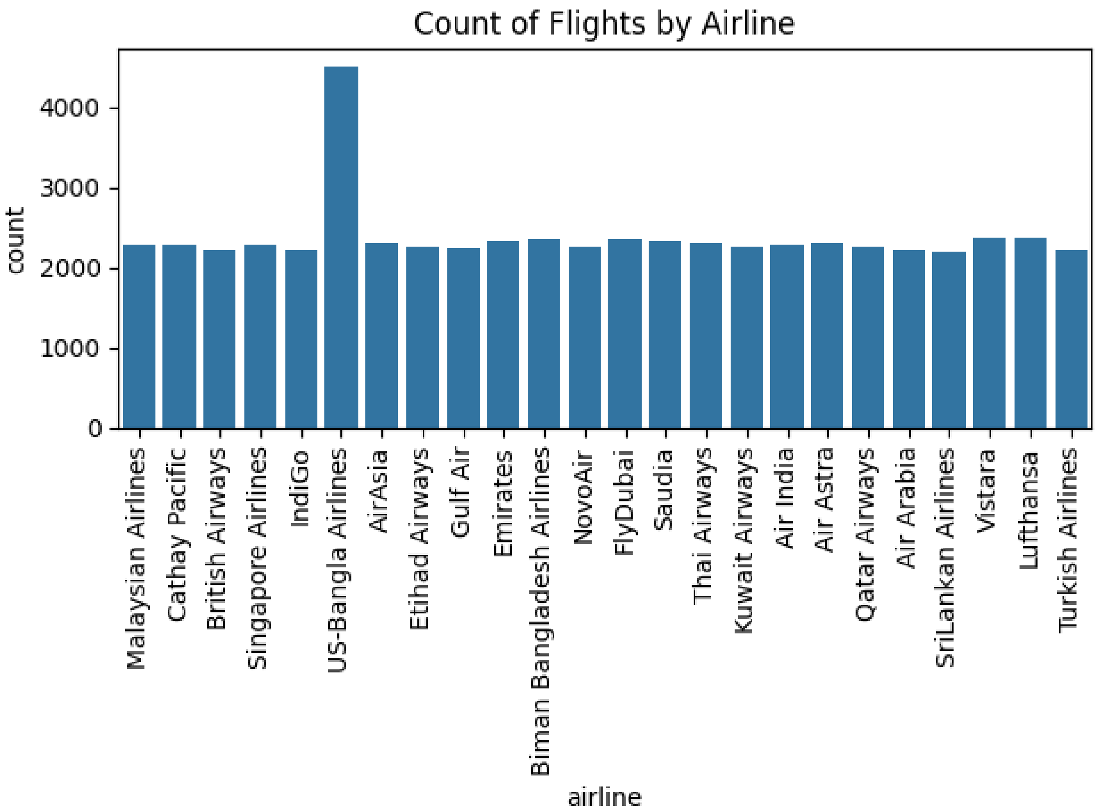
[ ]:

```
# Plot using seaborn
sns.histplot(df['total fare (bdt)'], kde=True, color='skyblue',
  edgecolor='black')
plt.xlabel('Total Fare (BDT)')
plt.title('Distribution of Total Fare')
plt.tight_layout()
plt.show()
```



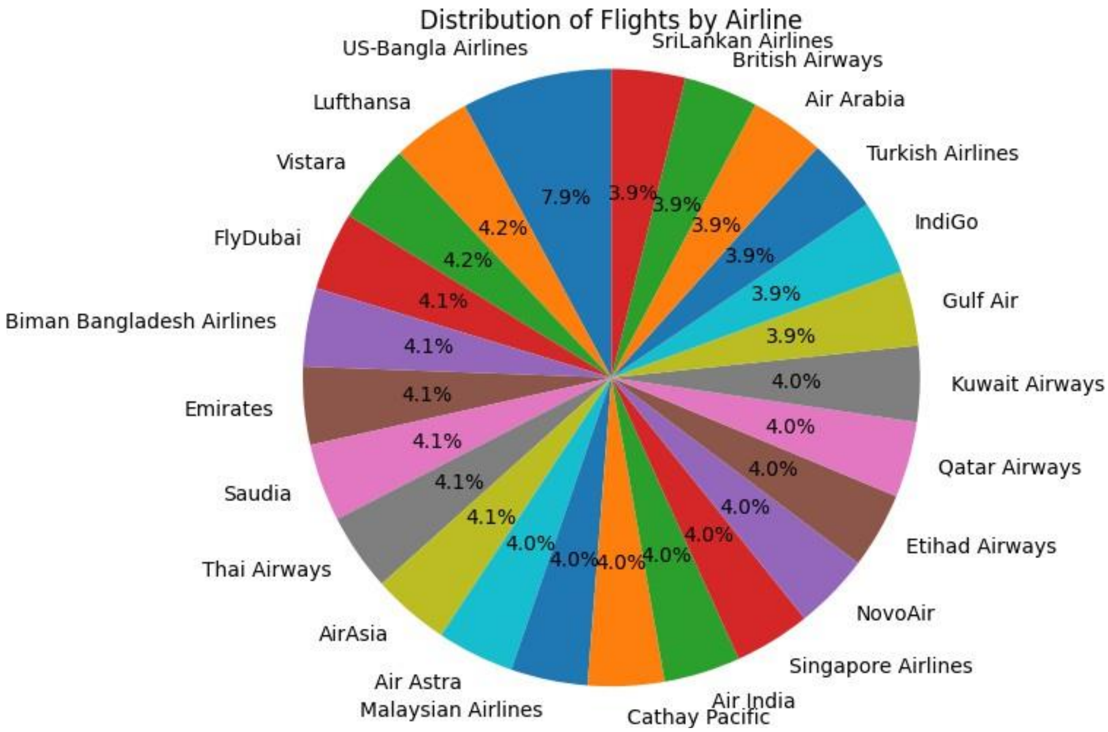
Bar Chart

[ ]:
```
# Use a valid categorical column like 'airline'
sns.countplot(x='airline', data=df)

plt.title('Count of Flights by Airline')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



Pie Chart

[ ]:
```
x = df['airline'].value_counts()

# Plot pie chart
plt.figure(figsize=(6, 6))
plt.pie(x.values, labels=x.index, autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Flights by Airline')
plt.axis('equal')  # Equal aspect ratio ensures the pie chart is circular
plt.show()
```

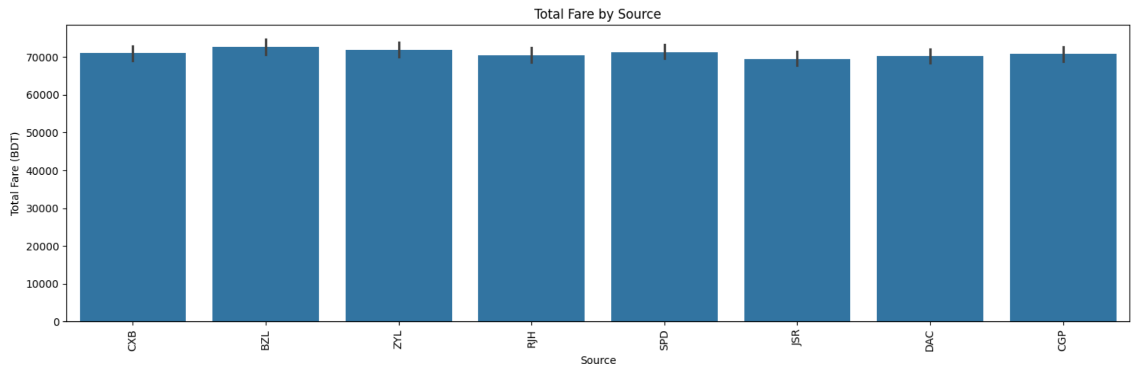


0.2 Bivariate analysis

Bivariate analysis is the simultaneous analysis of two variables. It explores the concept of the relationship between two variable whether there exists an association and the strength of this association or whether there are differences between two variables and the significance of these differences. The main three types we will see here are: 1. Categorical v/s Numerical 2. Numerical V/s Numerical 3. Categorical V/s Categorical dat

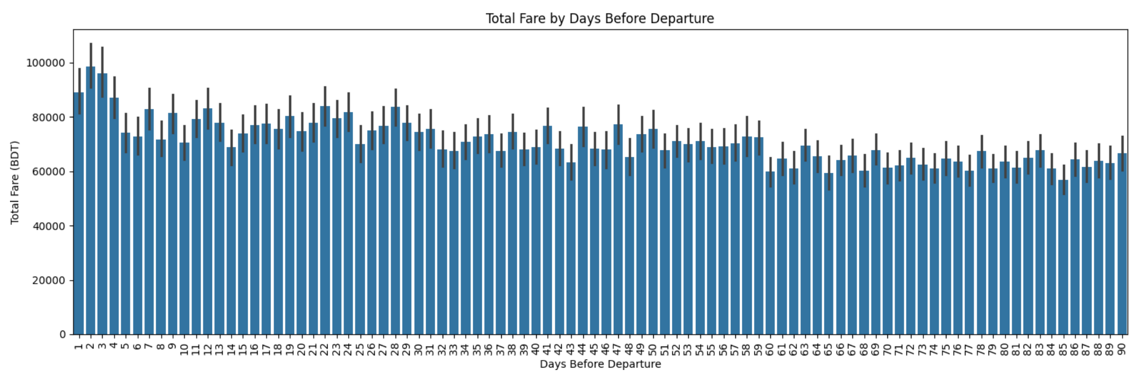
Categorical v/s Numerical

[ ]:
```
plt.figure(figsize=(15, 5))
sns.barplot(x='source', y='total fare (bdt)', data=df)
plt.xticks(rotation='vertical')
plt.title('Total Fare by Source')
plt.xlabel('Source')
plt.ylabel('Total Fare (BDT)')
plt.tight_layout()
plt.show()
```



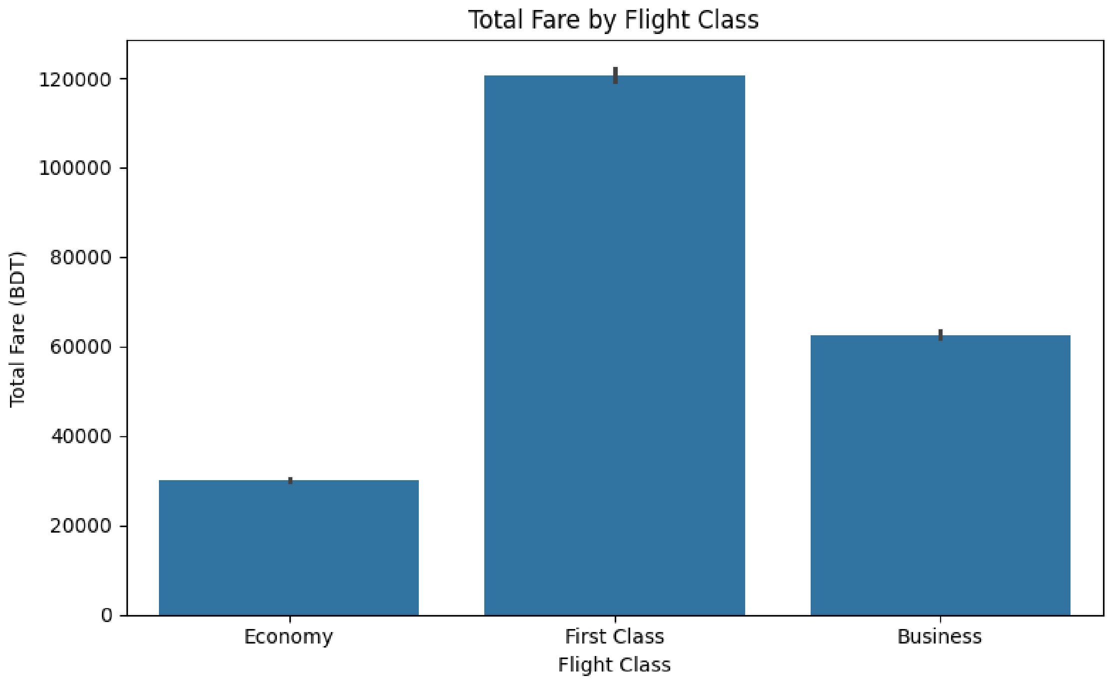
Numerical v/s Numerical

[ ]:
```
plt.figure(figsize=(15, 5))
sns.barplot(x='days before departure', y='total fare (bdt)', data=df)
plt.xticks(rotation='vertical')
plt.title('Total Fare by Days Before Departure')
plt.xlabel('Days Before Departure')
plt.ylabel('Total  Fare  (BDT)')
plt.tight_layout()
plt.show()
```



Categorical v/s Categorical

[ ]:
```
plt.figure(figsize=(8, 5))
sns.barplot(x='class', y='total fare (bdt)', data=df)
plt.title('Total Fare by Flight Class')
plt.xlabel('Flight Class')
plt.ylabel('Total Fare (BDT)')
plt.tight_layout()
plt.show()
```



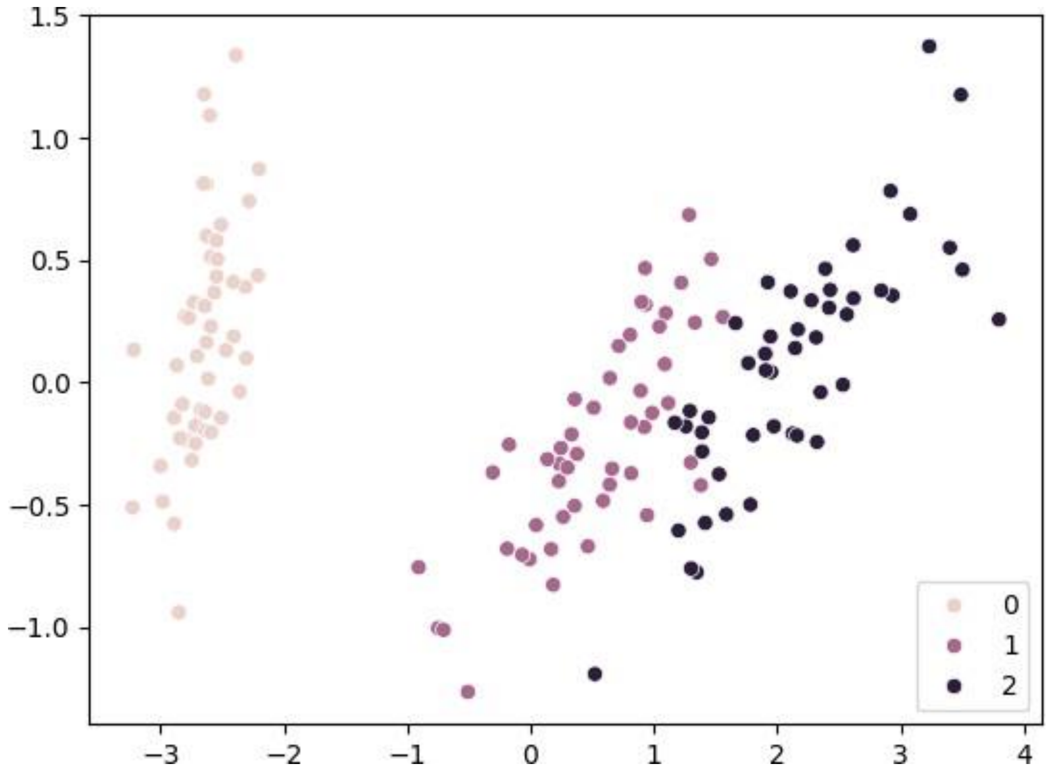
0.3 Multivariate Analysis

It is an extension of bivariate analysis which means it involves multiple variables at the same time to find correlation between them. Multivariate Analysis is a set of statistical model that examine patterns in multidimensional data by considering at once, several data variable.

PCA(Principal Component Analysis) PCA is a dimensionality reduction technique used in multivariate analysis. It reduces the number of variables while keeping the most important information. Why Use PCA? • Datasets with many variables can be complex and redundant. • PCA helps simplify the dataset by transforming it into fewer dimensions. • Helps in visualizing high-dimensional data in 2D or 3D plots.

[ ]:
```
from sklearn import datasets, decomposition
iris = datasets.load_iris()
X = iris.data
y = iris.target
pca = decomposition.PCA(n_components=2)
X = pca.fit_transform(X)
sns.scatterplot(x=X[:, 0], y=X[:, 1], hue=y)
```

[ ]: <Axes: >


1.5
1.0
0.5
0.0
−0.5
−1.0
−3
−2
−1
0
1
2
3
4
0
1
2

[ ]: