Problem Statement: Airline Price Prediction

Develop a predictive model to estimate airline ticket prices based on features such as airline, source, destination, flight duration, and total stops. Utilize machine learning techniques to create an accurate pricing model, providing valuable insights into the factors influencing airfare. The objective is to assist both customers in budget planning and airlines in setting competitive pricing strategies, enhancing overall transparency in the aviation industry.

Dataset Information:

-Airline: Categorical variable representing the airline of the flight. -Date_of_Journey: Date when the journey is scheduled to occur. -Source: Categorical variable indicating the departure location. -Destination: Categorical variable indicating the arrival location. -Route: String indicating the flight route. -Dep_Time: Time of departure for the flight. -Arrival_Time: Time of arrival for the flight. -Duration: Duration of the flight. -total_Stops: Categorical variable indicating the number of stops during the flight. -Additional_Info: Additional information about the flight. -Price: Numeric variable representing the ticket price.

Key Observations:

The dataset contains 10,682 entries with 11 columns.

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from warnings import filterwarnings
filterwarnings('ignore')
```

In [ ]:
```
# Create Dataframe and Read the dataset using Pandas
df = pd.read_csv('flight_price.csv')
df.head()
```

Out[ ]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Du
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4

In [ ]:
```
df.shape
```

Out[ ]:
```
(10683, 11)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10683 non-null  object
 1   Date_of_Journey  10683 non-null  object
 2   Source           10683 non-null  object
 3   Destination      10683 non-null  object
 4   Route            10682 non-null  object
 5   Dep_Time         10683 non-null  object
 6   Arrival_Time     10683 non-null  object
 7   Duration         10683 non-null  object
 8   Total_Stops      10682 non-null  object
 9   Additional_Info  10683 non-null  object
 10  Price            10683 non-null  int64 
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

In [ ]:
```
# Columns which has null values
```

```
df[df.isnull().any(axis=1)]
```

Out[ ]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	D
9039	Air India	6/05/2019	Delhi	Cochin	NaN	09:45	09:25 07 May	2

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

Out[ ]:
```
Airline           0
Date_of_Journey   0
Source            0
Destination       0
Route             1
Dep_Time          0
Arrival_Time      0
Duration          0
Total_Stops       1
Additional_Info   0
Price             0
dtype: int64
```

In [ ]:
```
# Remove null or na values rows
df =df.dropna().reset_index(drop=True)
df.shape
```

Out[ ]:
```
(10682, 11)
```

In [ ]:
```
# List out column names to check
df.columns
```

Out[ ]:
```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info', 'Price'],
      dtype='object')
```

In [ ]:
```
string_columns = df.select_dtypes(include='object').columns

# Check if there are any non-null values in those columns
columns_with_strings = [column for column in string_columns if df[column].notnul

print("Columns with string values:", columns_with_strings)
```

```
Columns with string values: ['Airline', 'Date_of_Journey', 'Source', 'Destinatio
n', 'Route', 'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops', 'Additional_I
nfo']
```

In [ ]:
```
# Spaces were fixed in the column names
df.columns = df.columns.str.strip()
df.columns
```

Out[ ]:
```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info', 'Price'],
      dtype='object')
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10682 entries, 0 to 10681
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          10682 non-null  object
 1   Date_of_Journey  10682 non-null  object
 2   Source           10682 non-null  object
 3   Destination      10682 non-null  object
 4   Route            10682 non-null  object
 5   Dep_Time         10682 non-null  object
 6   Arrival_Time     10682 non-null  object
 7   Duration         10682 non-null  object
 8   Total_Stops      10682 non-null  object
 9   Additional_Info  10682 non-null  object
 10  Price            10682 non-null  int64 
dtypes: int64(1), object(10)
memory usage: 918.1+ KB
```

In [ ]:
```
df.describe().T
```

Out[ ]:

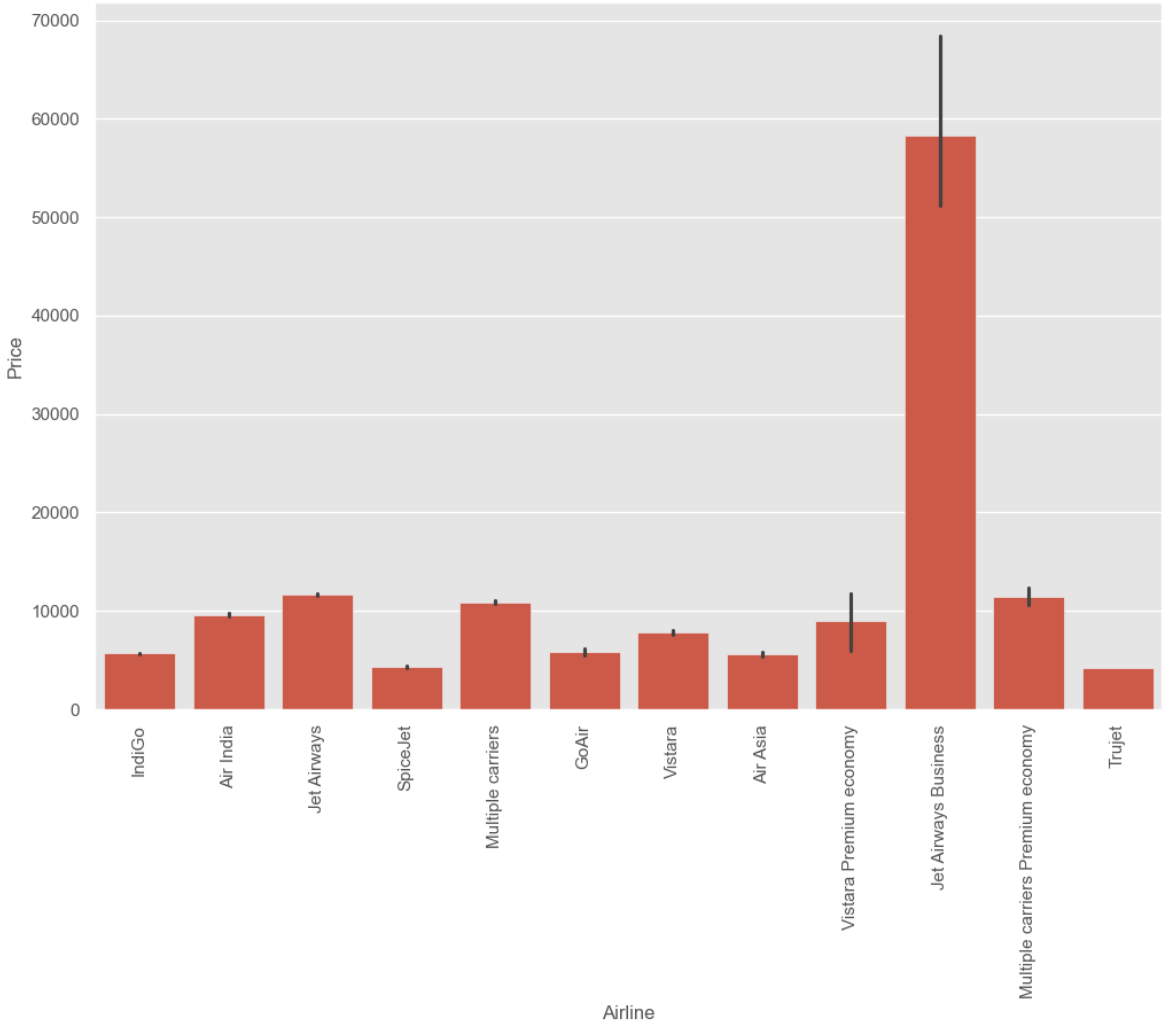
	count	mean	std	min	25%	50%	75%	max
Price	10682.0	9087.214567	4611.54881	1759.0	5277.0	8372.0	12373.0	79512.0

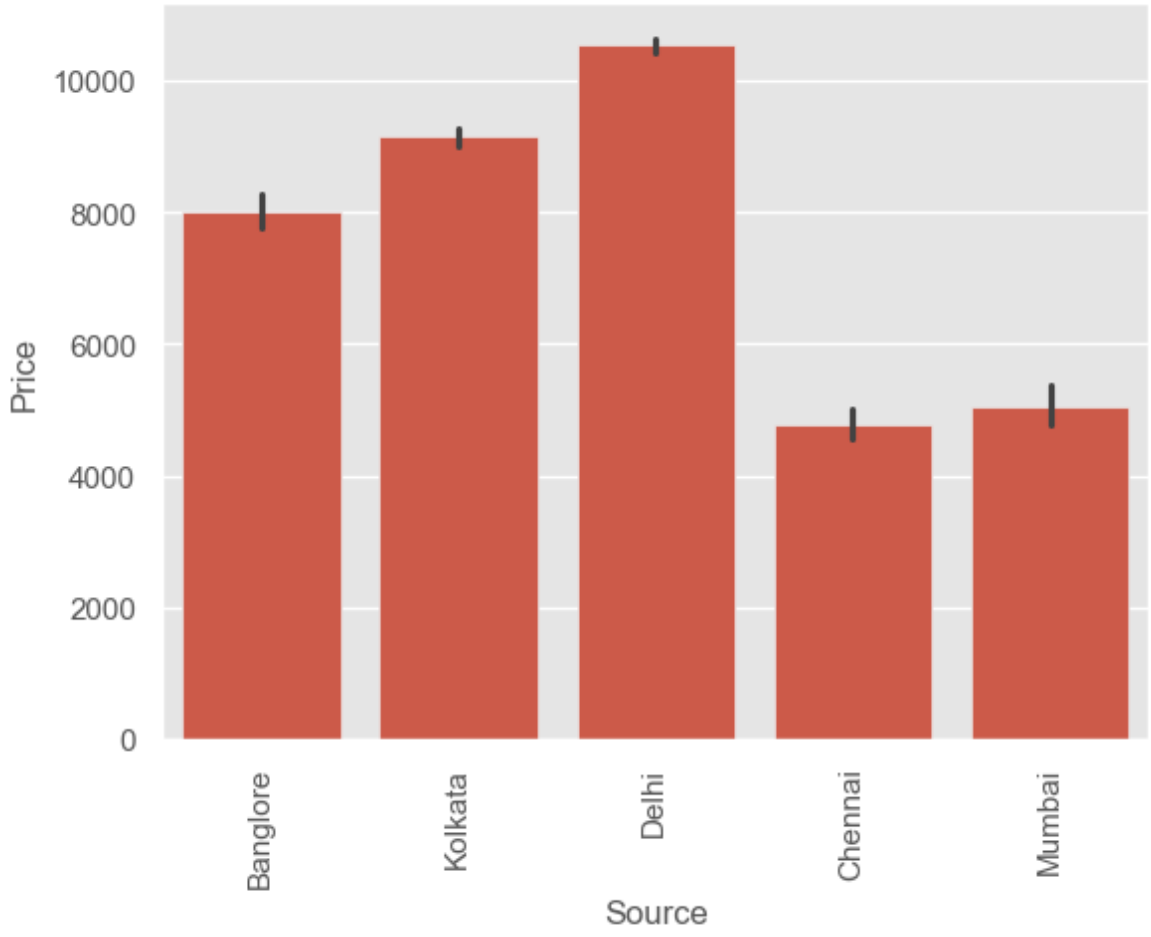
In [ ]:
```

```

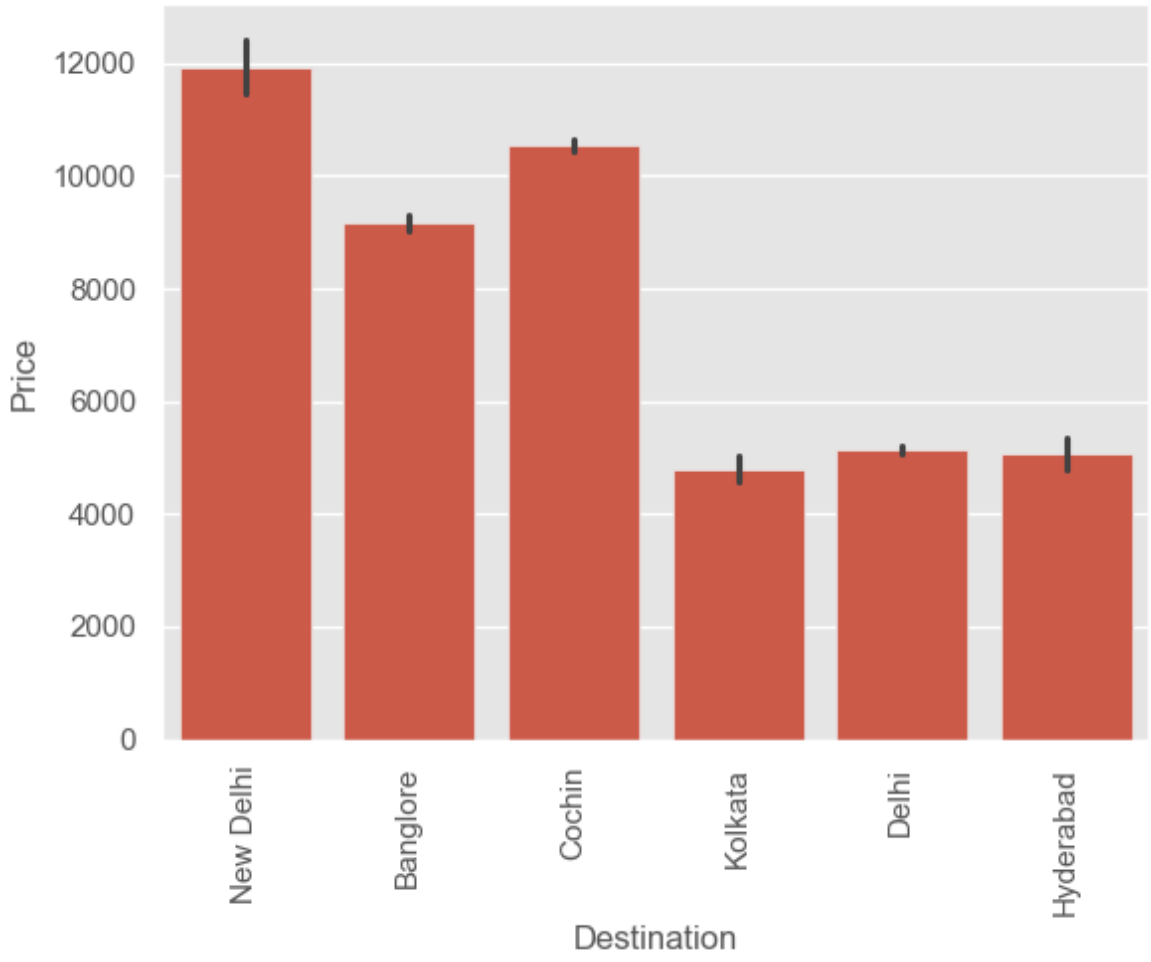
3.5 Exploratory Data Analysis (EDA)

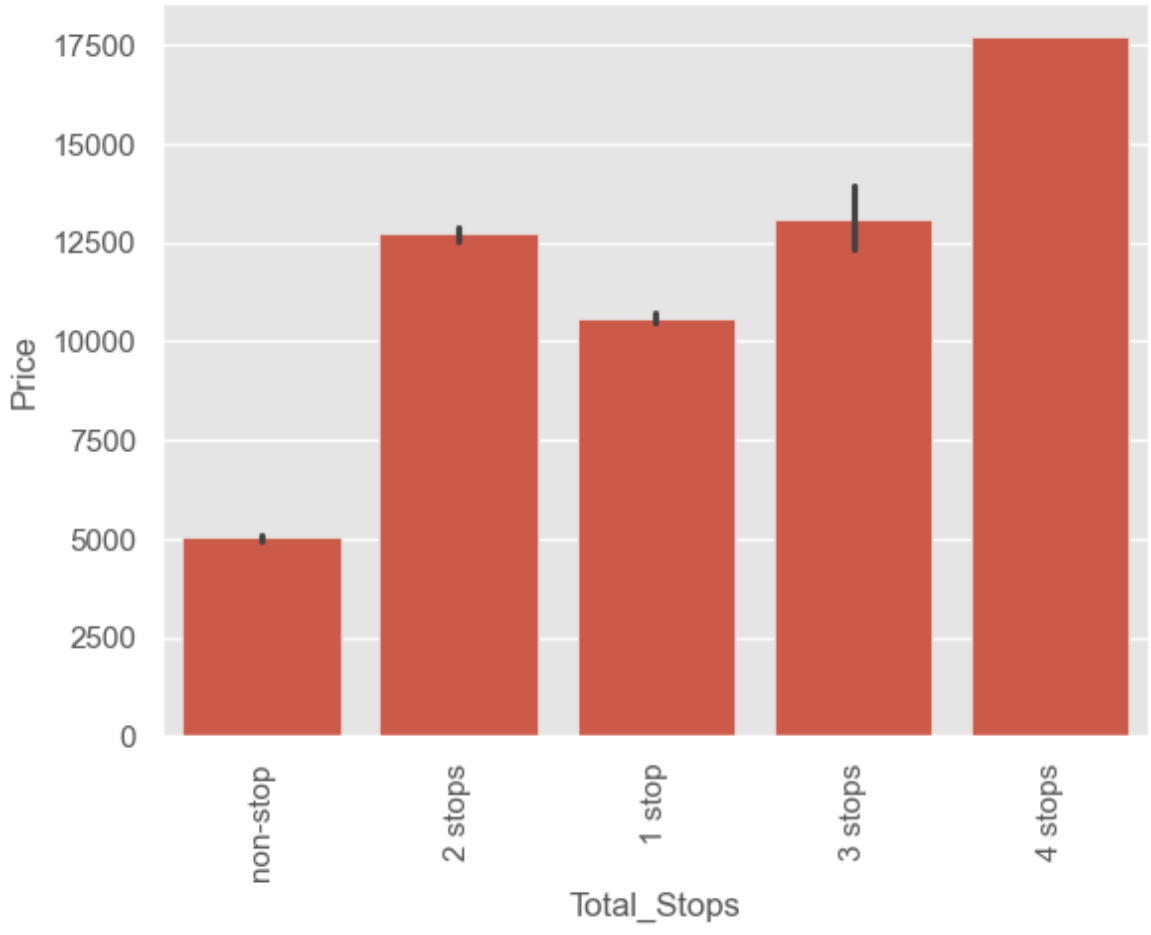
In [ ]:
```
#Droping Year features
df1 = df.drop(['Route', 'Dep_Time', 'Arrival_Time'], axis=1)
```

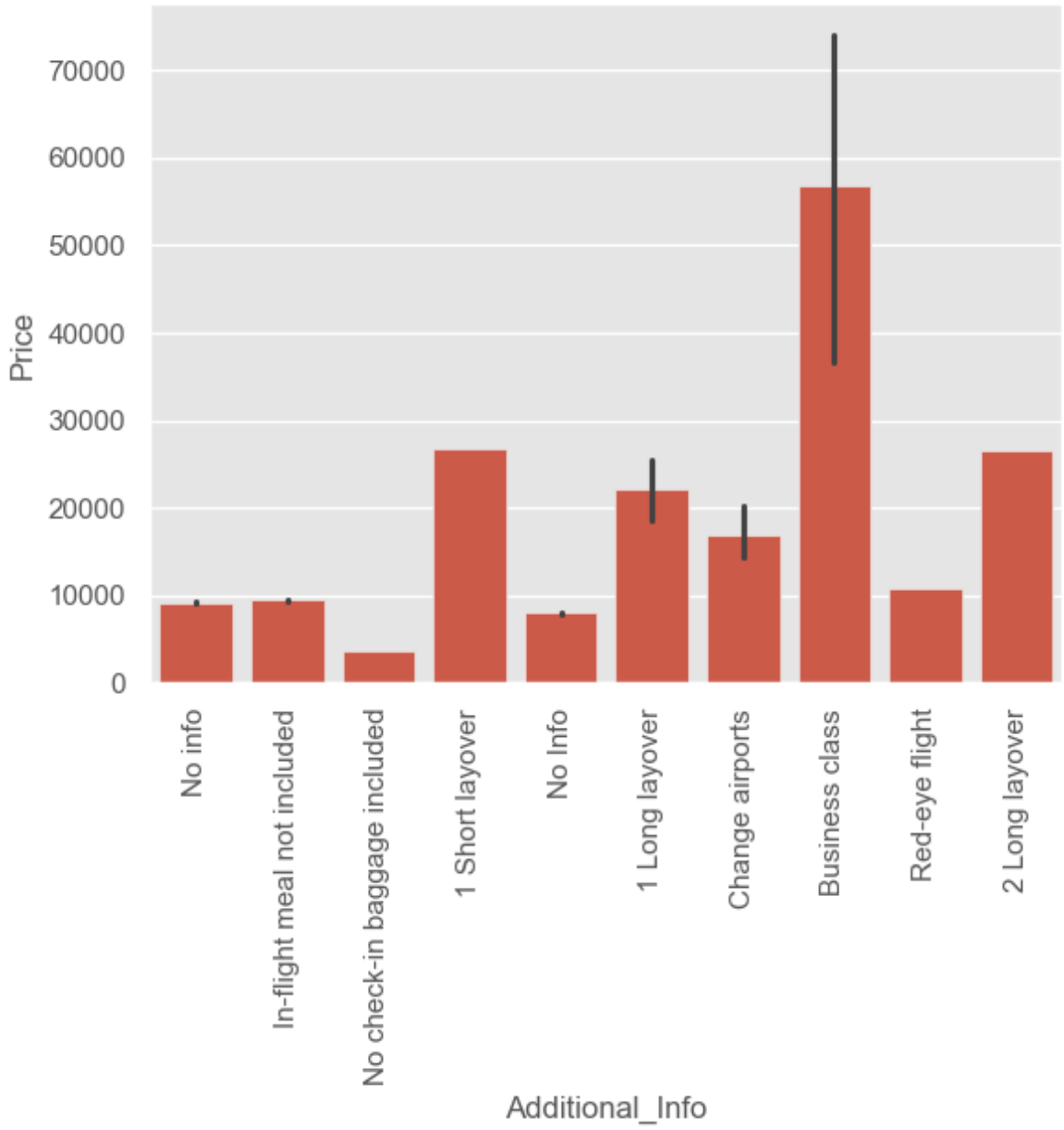
In [ ]:
```
plt.figure(figsize=(12, 8))

# Bar plots for categorical features
categorical_features = ['Airline', 'Source', 'Destination', 'Total_Stops', 'Addi
for feature in categorical_features:
    sns.barplot(x=feature, y='Price', data=df)
    plt.xticks(rotation=90)
    plt.show()


plt.show()
```


70000
60000
50000
40000
30000
20000
10000
0
Price
IndiGo
Air India
Jet Airways
SpiceJet
Multiple carriers
GoAir
Vistara
Air Asia
Vistara Premium economy
Jet Airways Business
Multiple carriers Premium economy
Trujet
Airline


10000
8000
6000
4000
2000
0
Price
Banglore
Kolkata
Delhi
Chennai
Mumbai
Source


12000
10000
8000
6000
4000
2000
0
Price
New Delhi
Banglore
Cochin
Kolkata
Delhi
Hyderabad
Destination


17500
15000
12500
10000
7500
5000
2500
0
Price
non-stop
2 stops
1 stop
3 stops
4 stops
Total_Stops



Insights: Airline vs. Price:

The bar plot for 'Airline' reveals the distribution of prices among different airlines. Insights: Identify airlines with higher or lower average prices. This information can help travelers choose more budget-friendly airlines. Source vs. Price:

The bar plot for 'Source' displays how the source location affects flight prices. Insights: Observe variations in prices based on the departure location. Some sources may be associated with higher or lower average prices. Destination vs. Price:

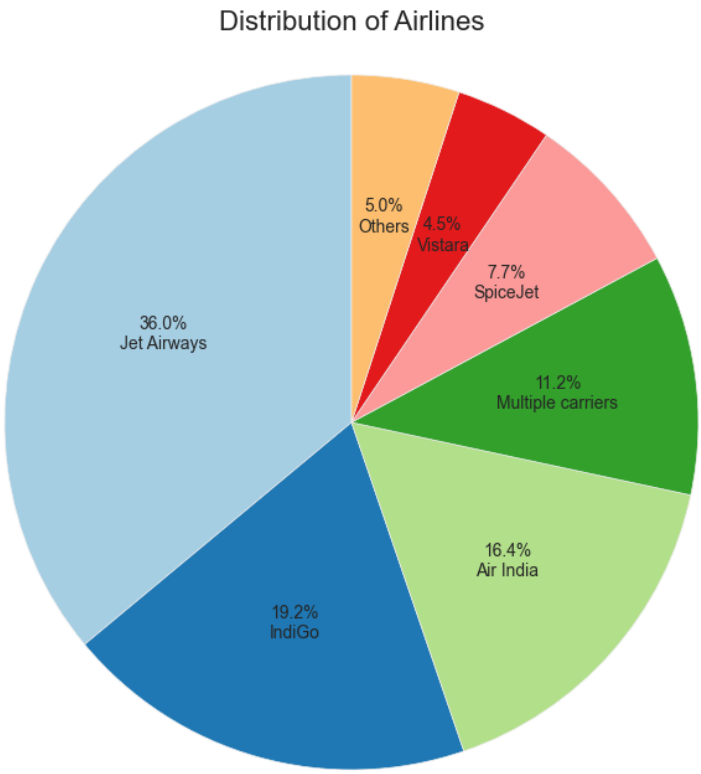
The bar plot for 'Destination' provides insights into the impact of destination on flight prices. Insights: Identify destinations with higher or lower average prices. This information can be useful for travelers planning their trips on a budget. Total_Stops vs. Price:

The bar plot for 'Total_Stops' illustrates how the number of stops influences flight prices. Insights: Understand the price differences between non-stop flights and those with

multiple stops. Travelers might use this information to make decisions based on their preferences and budget constraints. Additional_Info vs. Price:

The bar plot for 'Additional_Info' explores the impact of additional information on flight prices. Insights: Check if certain additional information leads to variations in prices. This could include factors such as in-flight services or special considerations that may affect pricing. General Observations: The plt.xticks(rotation=45) ensures that the x-axis labels are rotated for better readability, especially when dealing with categorical features with longer names. The loop iterates through each categorical feature, generating individual bar plots for each one. Conclusion: By examining these bar plots, you can gain valuable insights into how different categorical features influence flight prices. This information can be crucial for both travelers and airlines to make informed decisions related to ticket pricing and customer preferences.

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# Assuming df is your DataFrame
airline_counts = df['Airline'].value_counts()

plt.figure(figsize=(12, 8))

# Set a threshold for including airlines in the pie chart
threshold = 4.0  # You can adjust this threshold based on your preferences

# Filter out airlines with percentages below the threshold
included_airlines = airline_counts[airline_counts / sum(airline_counts) * 100 >=
excluded_airlines_count = sum(airline_counts) - sum(included_airlines)
included_airlines['Others'] = excluded_airlines_count

# Plot the pie chart without explode values
patches, texts, autotexts = plt.pie(included_airlines, labels=None, autopct='%1.

# Set labels for included airlines and 'Others'
for label, percent, autotext in zip(included_airlines.index, included_airlines /
    autotext.set_text(f'{percent:.1f}%\n{label}')

plt.title('Distribution of Airlines', fontsize=16)
plt.axis('equal')  # Equal aspect ratio ensures that the pie is drawn as a circl

plt.show()
```



Overview

This report presents an analysis of airline distribution in a dataset containing information about flight prices. The primary focus is on the distribution of different airlines and their impact on flight prices. The analysis includes a pie chart visualizing the market share of major airlines, with a specific emphasis on Jet Airways, IndiGo, Air India, Multiple Carriers, SpiceJet, Vistara, and other smaller carriers.

Methodology

The dataset contains various columns, including 'Airline' and 'Price.' The analysis is conducted by calculating the market share percentage for each airline based on the number of flights they operate.

Findings

The pie chart below illustrates the distribution of market share among different airlines:

Key Findings:

Jet Airways: 36.0% Jet Airways dominates the market with a significant share of 36.0%, indicating its popularity and potentially influencing flight prices.

IndiGo: 19.2% IndiGo holds a substantial market share of 19.2%, making it one of the leading airlines in terms of the number of flights.

Air India: 16.4% Air India, a major player in the aviation industry, commands a market share of 16.4%.

Multiple Carriers: 11.2% Flights operated by multiple carriers collectively represent 11.2% of the market, offering passengers a variety of options.

SpiceJet: 7.7% SpiceJet, with a market share of 7.7%, contributes to the diverse airline landscape.

Vistara: 4.5% Vistara, although with a smaller market share of 4.5%, adds to the overall competitiveness in the market.

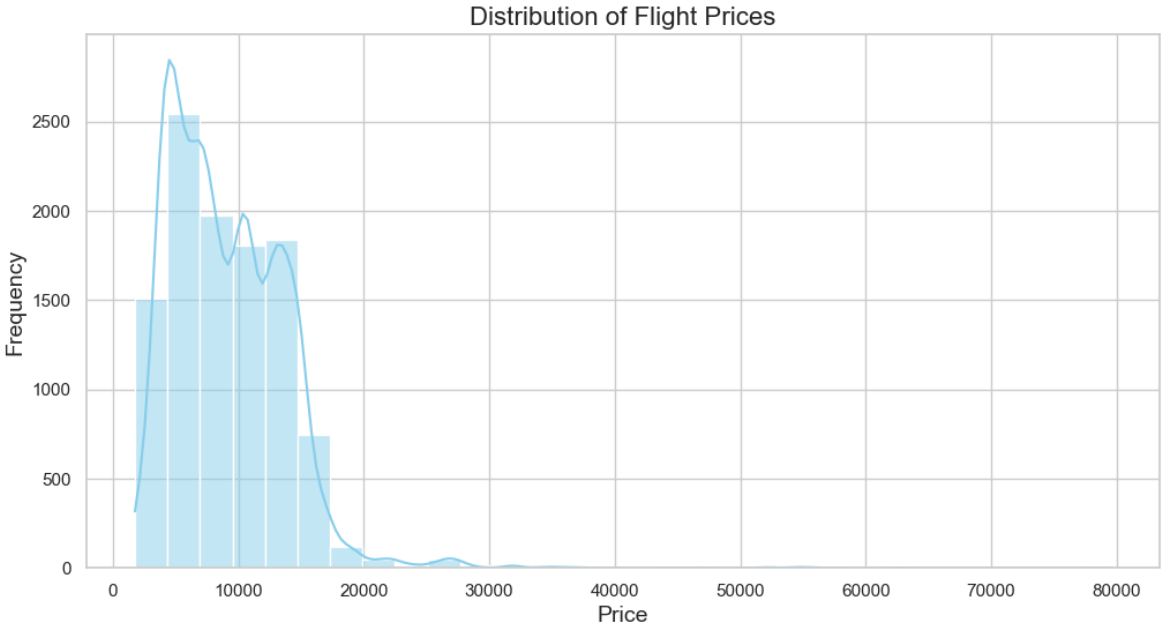
Others: 5.0% Smaller carriers collectively account for 5.0% of the market, showcasing the presence of various options for travelers.

INSIGHTS

The analysis of airline distribution provides valuable insights into the market dynamics, with Jet Airways emerging as a prominent player. The diversity in market share among airlines suggests a competitive landscape, influencing the pricing structure. Understanding the market share of each airline is crucial for both consumers and industry stakeholders in making informed decisions related to flight choices and pricing strategies.

This report serves as a foundation for further exploration into the factors affecting flight prices and the dynamics of the aviation industry.

In [ ]:
```
# Set the style for the plot
sns.set(style="whitegrid")

# Set the size of the plot
plt.figure(figsize=(12, 6))

# Plot the histogram for the 'Price' column
sns.histplot(df['Price'], bins=30, kde=True, color='skyblue')

# Set the title and labels
plt.title('Distribution of Flight Prices', fontsize=16)
plt.xlabel('Price', fontsize=14)
plt.ylabel('Frequency', fontsize=14)

# Show the plot
plt.show()
```



Report: Distribution of Flight Prices

Central Tendency:

The majority of flight prices seem to be concentrated in the lower range, with a peak around 10,000 to 15,000. There is a noticeable decrease in frequency as prices increase, suggesting that most flights are priced in the lower to mid-range. Outliers:

There are a few flights with prices significantly higher than the majority, indicating potential outliers or premium services. The long tail on the right side of the histogram suggests that there are some flights with considerably higher prices compared to the majority. Common Price Ranges:

A significant portion of flights falls within the range of 5,000 to 20,000, which is likely the common price range for various routes and airlines. Distribution Shape:

The histogram exhibits a slightly right-skewed distribution, indicating that while the majority of flights have lower prices, there are a few flights with higher-than-average prices. Insights:

Pricing Strategy:

Airlines may have a tiered pricing strategy with different fare classes, resulting in the observed peaks and variations in the histogram. The central concentration of prices may represent standard or economy class fares. Outliers Analysis:

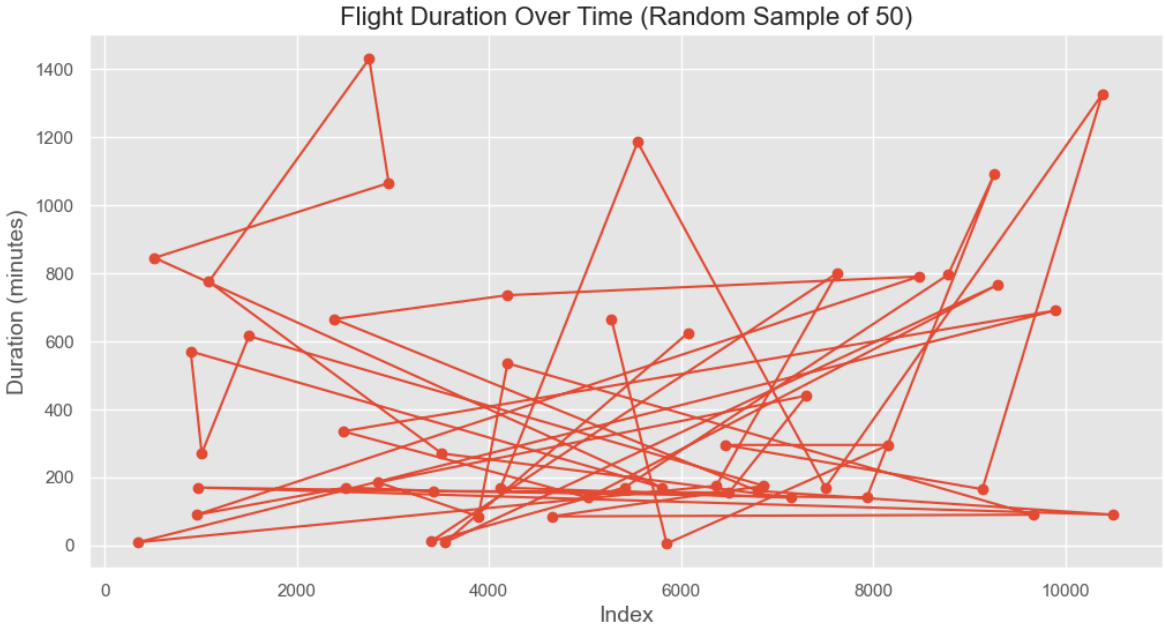
Further investigation is recommended for the flights with exceptionally high prices. Understanding the reasons behind these high prices can provide insights into premium services, peak travel times, or unique offerings. Customer Preferences:

Passengers may prefer more affordable flights, as suggested by the higher frequency in the lower price range. The distribution shape indicates that airlines may offer a variety of pricing options to cater to different customer segments. Market Dynamics:

This distribution can reflect market demand and supply dynamics. Prices are likely influenced by factors such as travel season, route popularity, and competition among airlines. Recommendations:

Airlines might consider optimizing pricing strategies based on the observed distribution to cater to the preferences of the majority of travelers. Further analysis, including correlation with other features, can provide a more comprehensive understanding of the factors influencing flight prices. This histogram offers a valuable overview of the pricing landscape, allowing for informed decision-making and strategy development within the airline industry.

In [ ]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Assuming the 'Duration' column is in the format 'Xh Ym'
# Convert the 'Duration' column to total minutes
df['Duration_minutes'] = df['Duration'].str.split().apply(lambda x: int(x[0][:-1

# Select a random sample of 50 rows
random_sample = df.sample(50, random_state=42)

# Set the style for the plot (you can change 'ggplot' to other styles)
plt.style.use('ggplot')

# Set the size of the plot
plt.figure(figsize=(12, 6))

# Plot the line plot for 'Duration_minutes' for the random sample
plt.plot(random_sample['Duration_minutes'], marker='o', linestyle='-')

# Set the title and labels
plt.title('Flight Duration Over Time (Random Sample of 50)', fontsize=16)
plt.xlabel('Index', fontsize=14)
plt.ylabel('Duration (minutes)', fontsize=14)

# Show the plot
plt.show()
```



Line Plot Analysis: Flight Duration Over Time

Objective: The line plot visualizes the duration of flights over time, focusing on a random sample of 50 data points from the 'Duration' column.

Insights:

Variability in Flight Durations:

The line plot reveals a noticeable variability in flight durations for the selected sample. Flight durations vary across different time points, indicating diverse travel times. Identifying Trends:

While the plot does not explicitly show a clear trend due to the random selection, it is possible to observe fluctuations in flight durations, suggesting that certain periods may experience longer or shorter flights. Outliers:

Outliers in flight duration may be identified as data points significantly deviating from the general pattern. However, without more specific time-related information, it's challenging to attribute these outliers to particular factors. Duration Format:

The 'Duration' column has been transformed into total minutes for ease of analysis. This allows for a more straightforward interpretation of flight durations. Recommendations:

Detailed Time Analysis:

For a more comprehensive understanding, consider conducting a detailed time-based analysis. This could involve grouping flights by specific time intervals or days of the week to identify patterns. Outlier Investigation:

Investigate the outliers in flight durations to determine if there are specific reasons for exceptionally long or short flights during certain periods. Additional Factors:

Integrate other relevant factors, such as airline, source, destination, or total stops, to explore potential correlations with flight durations. Data Enrichment:

Consider enriching the dataset with additional time-related information, such as the month or day of the week, to enable more nuanced temporal analyses. Conclusion: While the line plot provides a snapshot of flight durations for a random sample, further exploration with additional contextual information is recommended for a more comprehensive understanding of the underlying patterns and factors influencing flight durations.

Conclusion:

In conclusion, the dataset provides comprehensive information about flight details, including airlines, journey dates, source, destination, routes, departure and arrival times, flight duration, total stops, additional information, and ticket prices. Initial exploration indicates potential areas for analysis, such as understanding the impact of categorical variables on ticket prices and exploring temporal trends in flight schedules. Further preprocessing, cleaning, and statistical analyses will be essential to unveil insights and patterns within the dataset, facilitating informed decision-making for stakeholders in the aviation industry.