Exploratory Danalysis (EDA) Report

1. Introduction

This project involves performing Exploratory Data Analysis (EDA) on a dataset containing information about flights, such as airline names, flight numbers, departure and arrival cities, times, travel class, duration, and ticket prices. The primary goal is to analyze the relationships between various features, such as ticket price, airline, time, and location, and to visualize these insights.

---

2. Importing Libraries

We first import the necessary libraries for handling data and creating visualizations:

```
Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

Import seaborn as sns
```

---

3. Loading the Dataset

We load the dataset using the Pandas read_csv function:

```
Flight = pd.read_csv("D:\\flight data.csv")
```

After loading, we inspect the first and last 5 rows to get a glimpse of the data:

```
# Display first 5 rows

Flight.head()


# Display last 5 rows

Flight.tail()
```

---

4. **Checking for Missing Values**

To ensure the dataset has no missing values, we use the isnull()function to check null values in a dataset .sum() function null counts in each column

Flight.isnull().sum()

Key Insight: There are no missing values in the dataset.

---

5. **Data Overview**

We retrieve basic information about the data types and memory usage:

Flight.info()

To better understand the numerical columns, we generate a summary of statistical information:

Flight.describe().T

T is used to show statistical summary transpose .

For the categorical columns, we also generate statistical summaries:

Flight.describe(include='object')

---

6. **Data Cleaning and Preparation**

We remove the index column to clean up the data:

Removing the index column from display

Flight.head().style.hide(axis='index')

---

7. Data Visualization and Insights

7.1. Airlines Count

We analyze the frequency of flights by each airline:

Airline = flight[‘airline’].value_counts()

Value_counts used for occurrences.

Plotting the count of flights by airlines

Plt.figure(figsize=(10, 6))

Sns.barplot(x=Airline.index, y=Airline.values, palette=’viridis’)

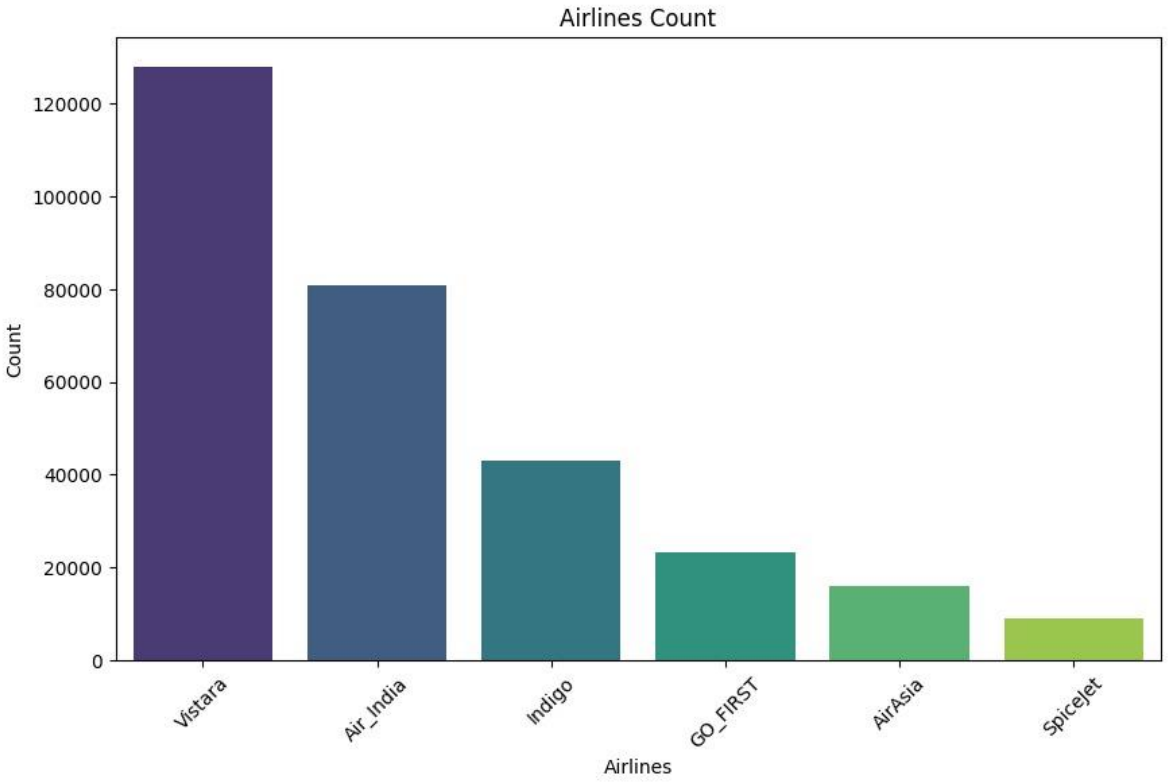
Plt.xticks(rotation=45)

Plt.title(“Airlines Count”)

Plt.xlabel(“Airlines”)

Plt.ylabel(“Count”)

Plt.show()



Vistara operates the highest number of flights, followed by Air India.

7.2. Departure Time vs. Arrival Time

We explore the relationship between departure and arrival times:

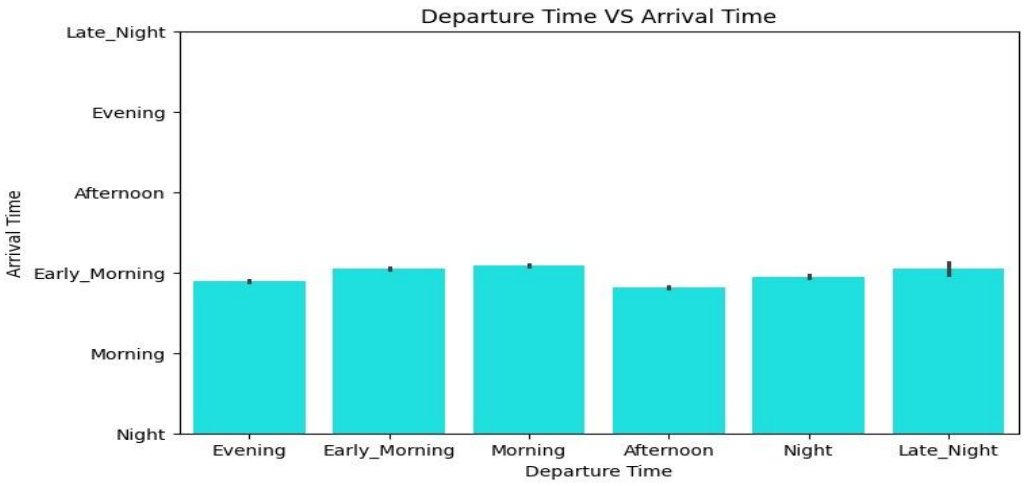
Plt.figure(figsize=(8, 5))

Sns.barplot(x='departure_time', y='arrival_time', data=flight, color='cyan')

Plt.title('Departure Time vs Arrival Time')

Plt.xlabel('Departure Time')

Plt.ylabel('Arrival Time')



The plot helps visualize the most common times for departures and arrivals.

7.3. Source City vs. Destination City

We examine the distribution of flights between various cities:

Data = flight.groupby([‘source_city’, ‘destination_city’]).size().unstack()

Plotting the stacked bar chart

Data.plot(kind=’bar’, stacked=True, figsize=(10, 6), colormap=’plasma’)

Plt.title(‘Source City vs Destination City’)

Plt.xlabel(‘Source City’)

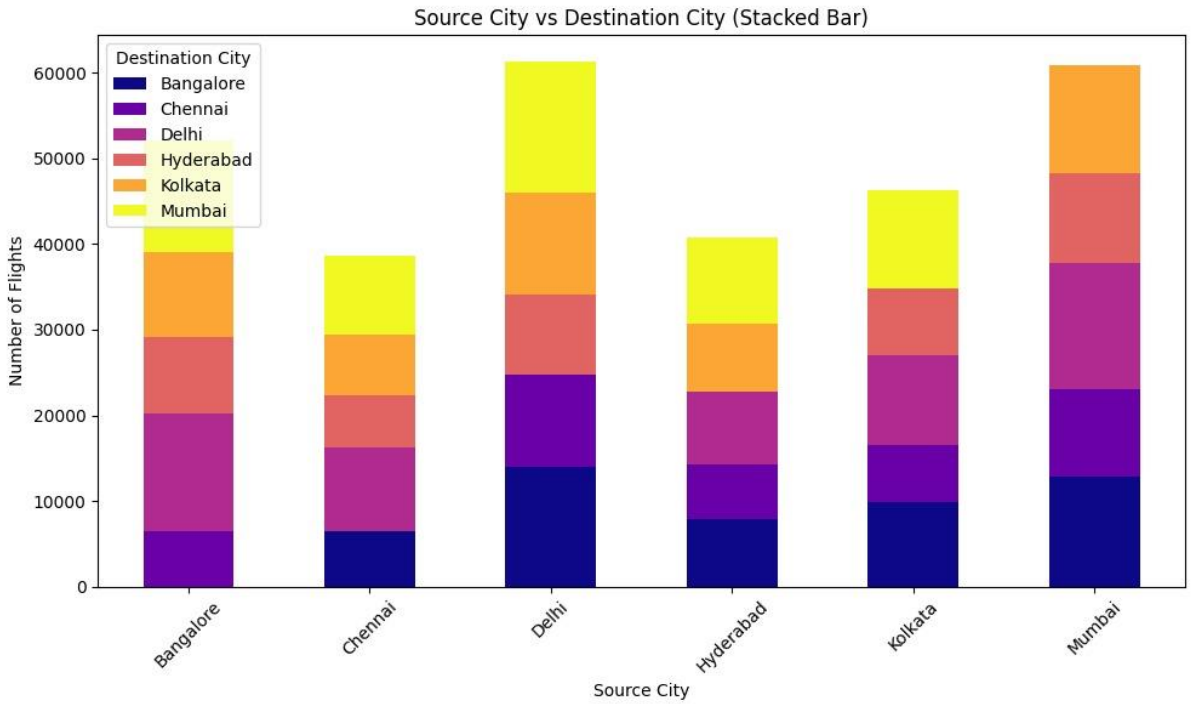
Plt.ylabel(‘Number of Flights’)

Plt.xticks(rotation=45)

Plt.legend(title=’Destination City’)

Plt.tight_layout()

Plt.show()



Popular routes include flights between Delhi and Mumbai, and Chennai and Hyderabad.

7.4. Price Variance Across Airlines

We analyze how ticket prices vary among airlines:

Plt.figure(figsize=(8, 5))

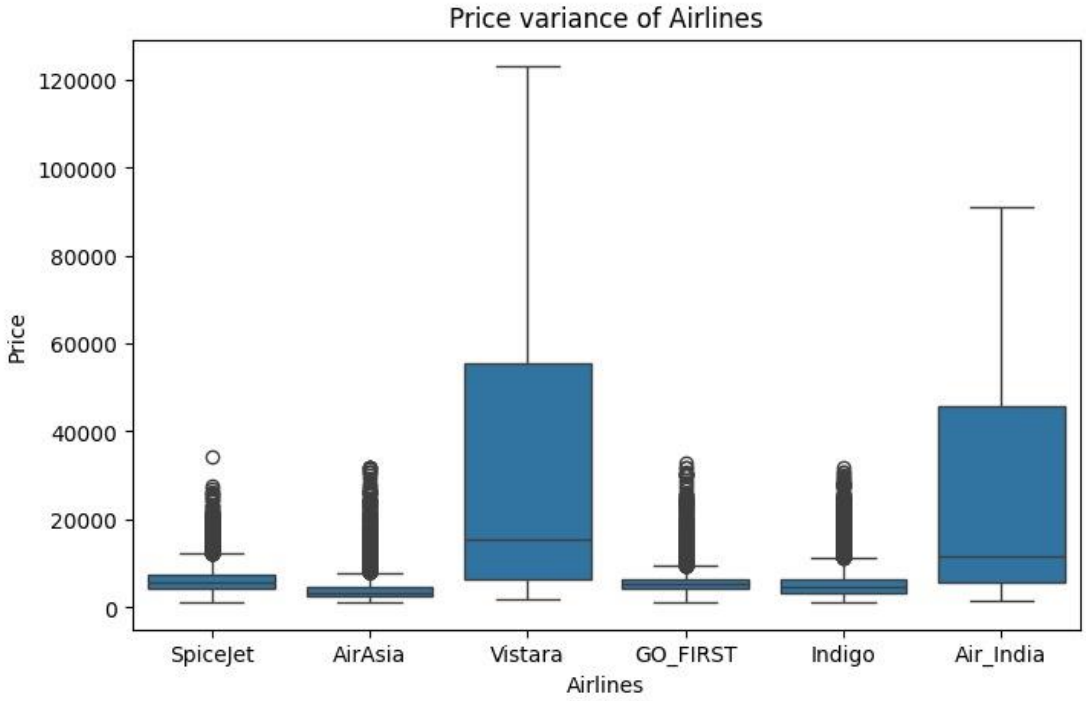
Sns.boxplot(x='airline', y='price', data=flight)

Plt.title('Price Variance Across Airlines')

Plt.xlabel('Airlines')

Plt.ylabel('Price')

Plt.show()



Airlines like Air India and Vistara have a higher price range than others.

7.5. Price Variance After Log Transformation

To handle outliers, we apply a log transformation to ticket prices:

Flight[‘price’] = np.log1p(flight[‘price’])

Plt.figure(figsize=(8, 5))

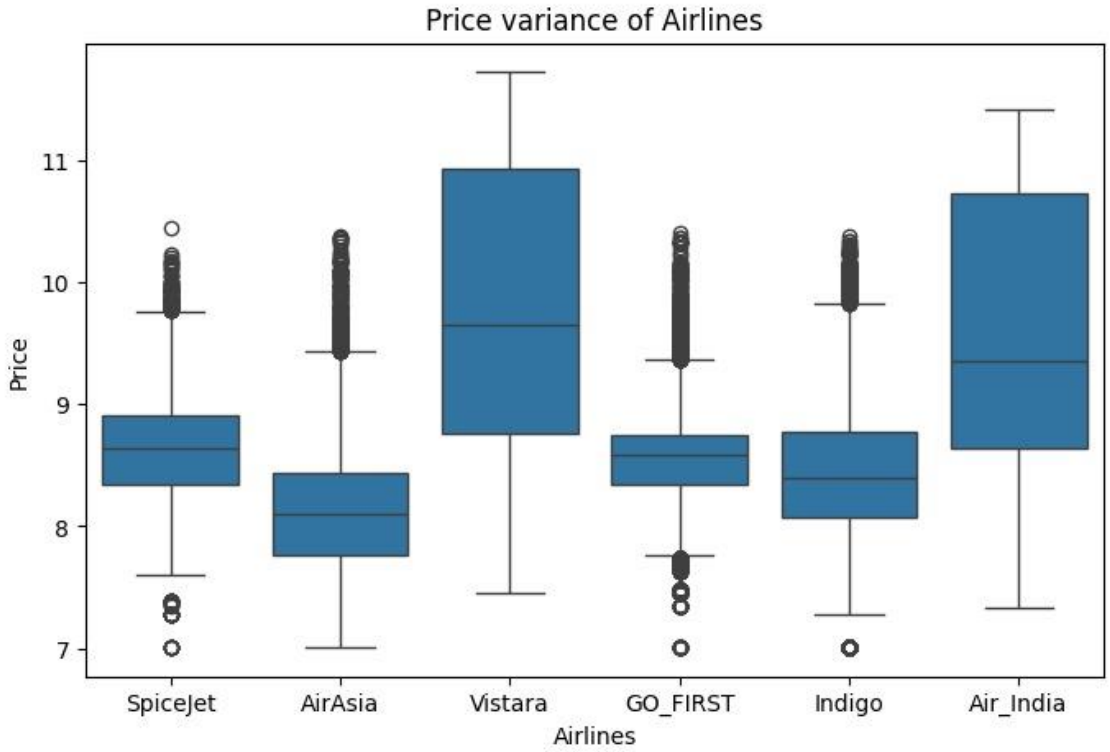
Sns.boxplot(x=’airline’, y=’price’, data=flight)

Plt.title(‘Price Variance of Airlines (After Log Transformation)’)

Plt.xlabel(‘Airlines’)

Plt.ylabel(‘Log of Price’)

Plt.show()



The log transformation reduces skewness caused by outliers, making price distributions more uniform.

7.6. Price Based on Departure and Arrival Times

We explore how ticket prices vary with departure and arrival times:

```
Plt.figure(figsize=(12, 8))

# Plot for departure time

Plt.subplot(2, 1, 1)

Plt.plot(flight['departure_time'], flight['price'], marker='o', color='blue')

Plt.title('Ticket Price by Departure Time')

Plt.xlabel('Departure Time')

Plt.ylabel('Log of Price')


# Plot for arrival time
```

Plt.subplot(2, 1, 2)

Plt.plot(flight[‘arrival_time’], flight[‘price’], marker=’o’, color=’green’)

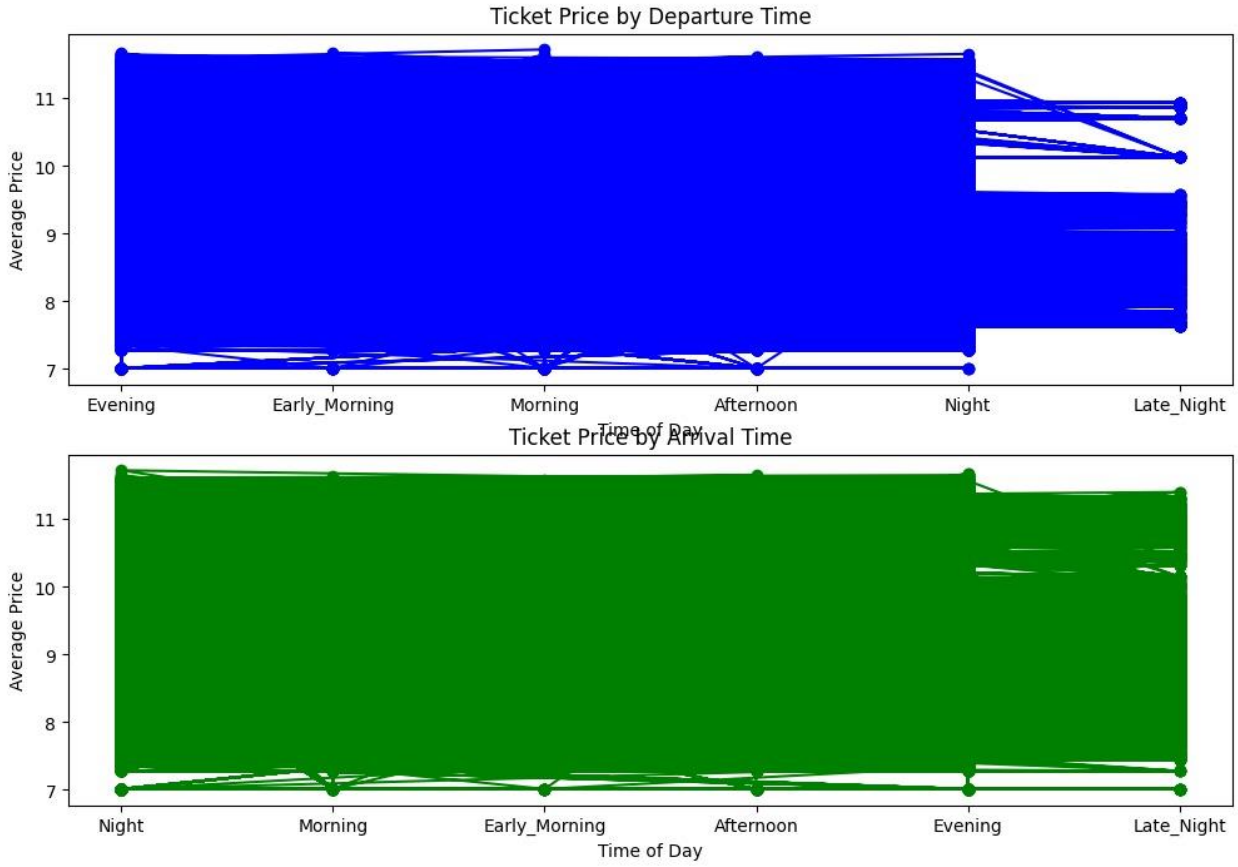
Plt.title(‘Ticket Price by Arrival Time’)

Plt.xlabel(‘Arrival Time’)

Plt.ylabel(‘Log of Price’)

Plt.tight_layout()

Plt.show()



Fliights during Morning and Evening times tend to have higher ticket prices.

7.7. Price Changes with Source and Destination

We use a heatmap to observe how ticket prices vary between different source and destination cities:

```
Aggregated = flight.groupby(['source_city', 'destination_city'])['price'].mean().reset_index()

# Pivot the table for heatmap

Pivot = aggregated.pivot(index='source_city', columns='destination_city', values='price')

# Plot the heatmap

Plt.figure(figsize=(12, 6))

Sns.heatmap(pivot, annot=True, cmap='coolwarm', fmt='.0f')

Plt.title('Price Changes by Source and Destination Cities')

Plt.show()
```

Routes like Delhi to Mumbai have higher average prices.

7.8. Duration of Travel vs. Cities

We analyze how the duration of flights varies by city pairs:

Mean_duration = flight.groupby([‘source_city’, ‘destination_city’])[‘duration’].mean().reset_index()

Pivot the table for heatmap

Duration_pivot = mean_duration.pivot(index=’source_city’, columns=’destination_city’, values=’duration’)

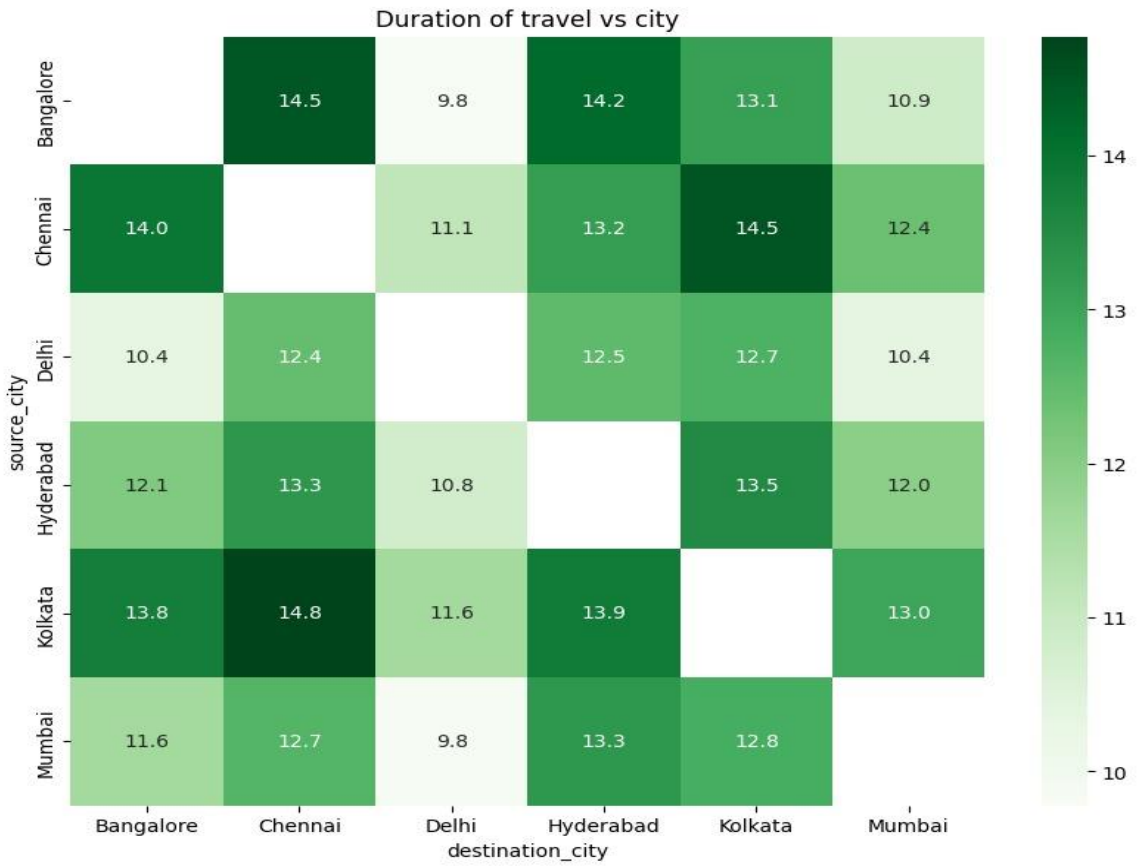
Plot the heatmap

Plt.figure(figsize=(10, 8))

Sns.heatmap(duration_pivot, annot=True, cmap=’Greens’, fmt=’.1f’)

Plt.title(“Duration of Travel vs. Cities”)

Plt.show()

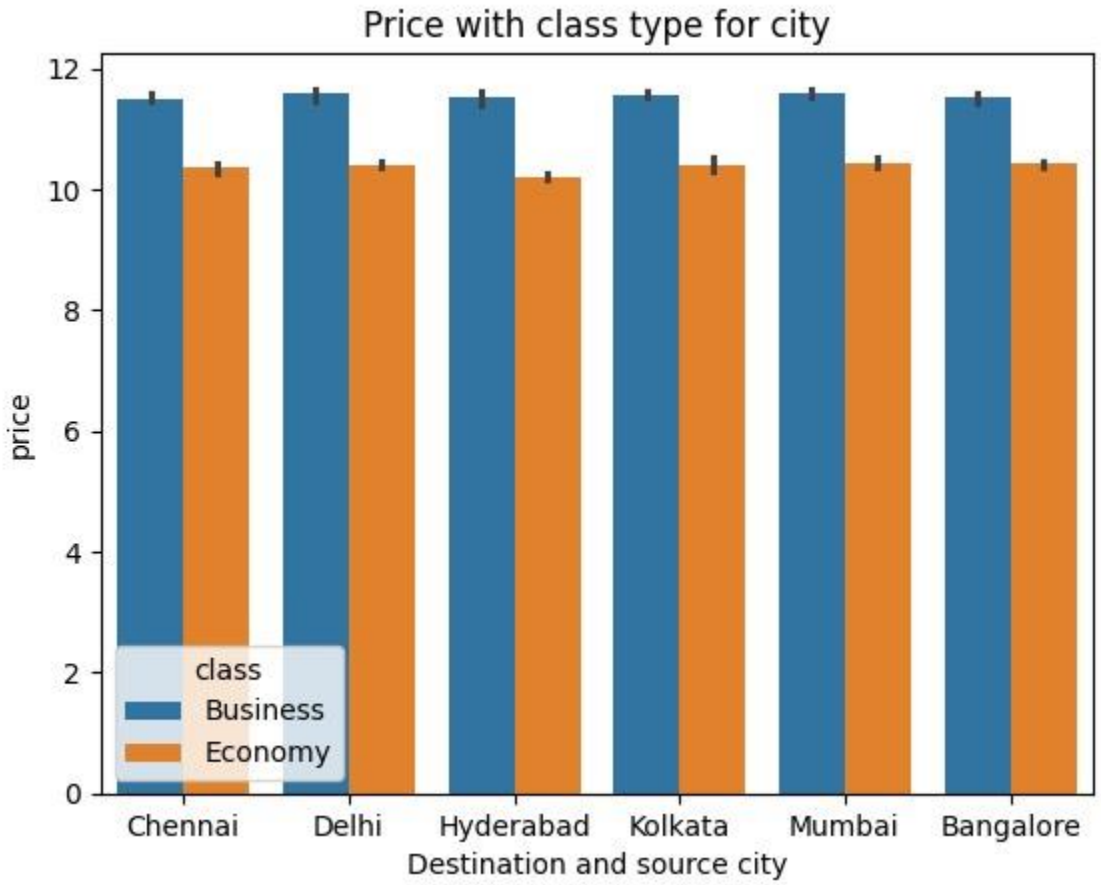


Fllights on routes like Delhi to Hyderabad have longer durations.

7.9. Price by Class and Cities

We compare prices between Economy and Business classes across different routes:

```
High_price = flight.groupby([‘source_city’, ‘destination_city’,
‘class’])[‘price’].max().reset_index()

# Plot the bar plot

Plt.figure(figsize=(12, 6))

Sns.barplot(x=’destination_city’, y=’price’, hue=’class’, data=high_price)

Plt.title(‘Price by Class Type and Cities’)

Plt.xlabel(‘Destination City’)

Plt.ylabel(‘Price’)

Plt.show()
```



Business class tickets are significantly more expensive than Economy class, especially for major cities.

---

8. Conclusion

This EDA provided useful insights into how various factors like airlines, departure/arrival times, and city routes affect flight prices and durations. Through visualizations, we observed patterns in ticket pricing, travel duration, and route frequency, which could help in price optimization for airlines and better travel planning for passengers.