Flight Price Prediction

Data Source: https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction

importing the primary libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

(1) **Data Loading**

```
flight_data=pd.read_csv('/content/drive/MyDrive/Clean_Dataset.csv')
```

```
# reading the 1st 3 rows of the dataset
flight_data.head(3)
```

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	desti
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	
			SG-					

As the column Unnmed: 0 is not needed, it is dropped

```
flight_data=flight_data.drop(columns=['Unnamed: 0'])
```

Reading the dataset

```
# reading the 1st 3 rows of the dataset
flight_data.head(3)
```

	airline	flight	source_city	departure_time	stops	arrival_time	destination_cit
0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumba
1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumba

```
# reading the last 3 rows of the dataset
flight_data.tail(3)
```

	airline	flight	source_city	departure_time	stops	arrival_time	destinatio
300150	Vistara	UK-832	Chennai	Early_Morning	one	Night	Hyc
300151	Vistara	UK-828	Chennai	Early_Morning	one	Evening	Hyc

(2) **Data Preprocessing**

Dimensions of the dataset

```
flight_data.shape
```

```
(300153, 11)
```

Checking the data types for each column

```
flight_data.dtypes
```

```
airline             object
flight              object
source_city         object
departure_time      object
stops               object
```

```
arrival_time         object
destination_city     object
class                object
duration            float64
days_left             int64
price                 int64
dtype: object
```

Checking for null, missing or duplicate values in the dataset.

```
print('Null values:',flight_data.isnull().any().sum())
print('NaN values:', flight_data.isna().any().sum())
print('duplicates:',flight_data.duplicated().any().sum())
```

```
Null values: 0
NaN values: 0
duplicates: 0
```

(3) **Exploratory Data Analysis**

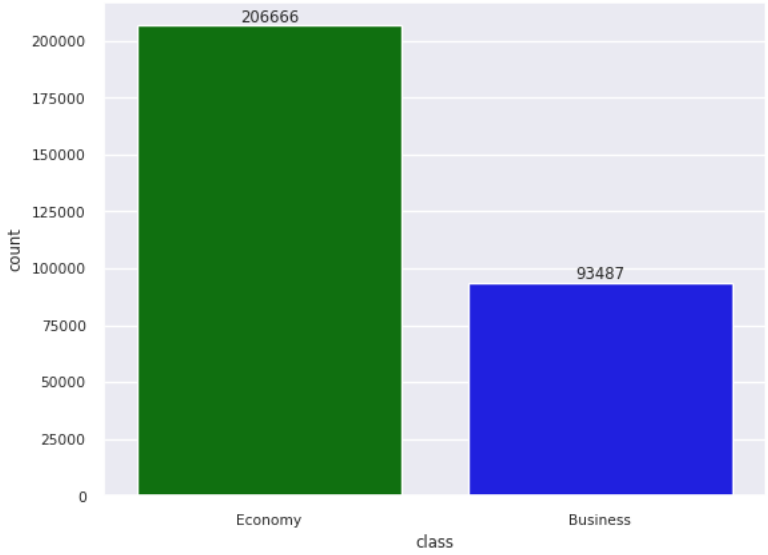
a. Checking for no.of distinct values in each column in the dataset

```
flight_data.nunique()
```

```
airline                 6
flight               1561
source_city             6
departure_time          6
stops                   3
arrival_time            6
destination_city        6
class                   2
duration              476
days_left              49
price               12157
dtype: int64
```

b. No.of flights per class - Economy and Business

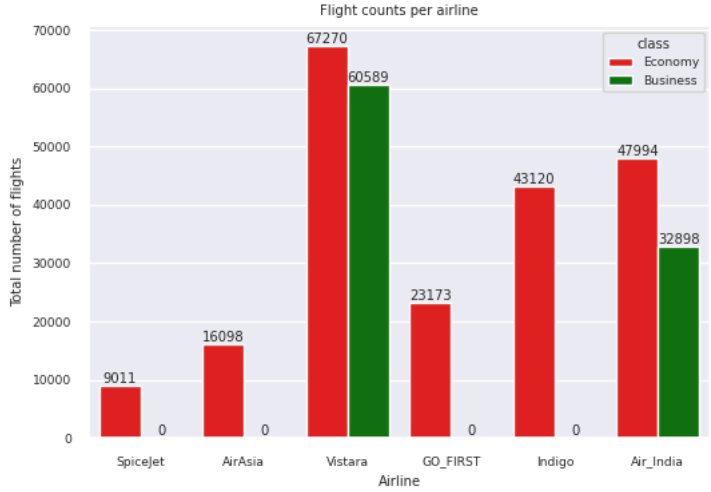
```
sns.set(font_scale=0.7)
cl={'Economy':'green','Business':'blue'}
c=sns.countplot(data=flight_data,x='class',palette=cl)
for label in c.containers:
  c.bar_label(label)
```



c. Total number of flights under each Airline and class

```
sns.set(font_scale=0.6)
plt.figure(figsize=(6,4))
col={'Economy':'red','Business':'green'}
a=sns.countplot(data=flight_data,x='airline',hue='class',palette=col)
for l in a.containers:
  a.bar_label(l)
plt.title('Flight counts per airline')
plt.xlabel('Airline')
plt.ylabel('Total number of flights')
```

```
Text(0, 0.5, 'Total number of flights')
```



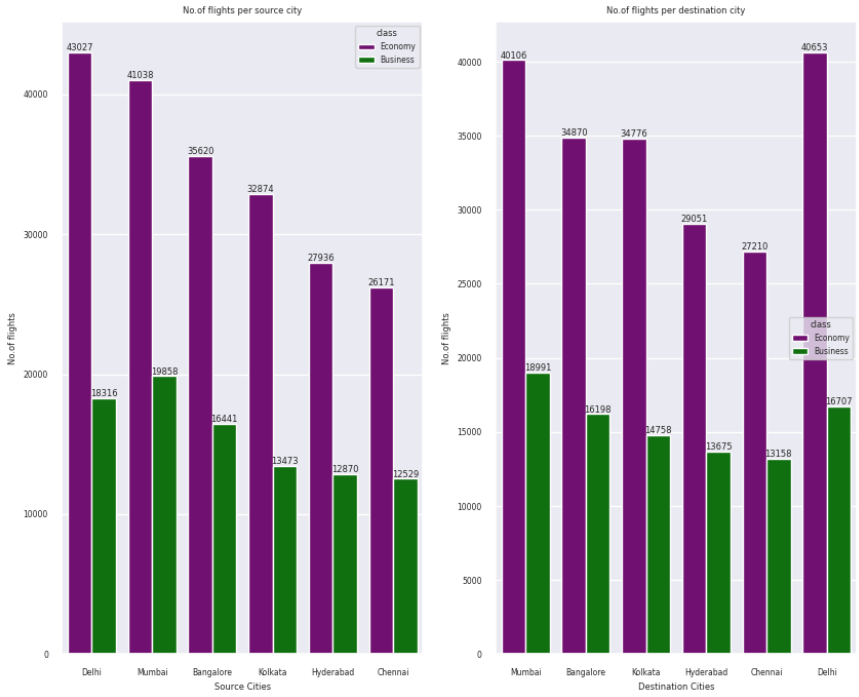
1. Among the six airlines, only Vistara and Air India have both classes Economy and Business
2. And the airline Vistara has the highest no.of flights from both classes
3. Spicejet is the airline which has lowest no.of flights

d. Plotting No.of flights per cities and class category

```
sns.set(font_scale=0.5) # setting the font scale
plt.figure(figsize=(10,8)) # setting the chart size

plt.subplot(1,2,1) # 1st plot in the subplot
col={'Economy':'purple','Business':'green'}
ax=sns.countplot(data=flight_data,x='source_city',hue='class',palette=col)
plt.title('No.of flights per source city')
plt.xlabel('Source Cities')
plt.ylabel('No.of flights')
for label in ax.containers:
    ax.bar_label(label) # adding label to the bars

plt.subplot(1,2,2) # 2nd plot in the sub plot
col={'Economy':'purple','Business':'green'}
bx=sns.countplot(data=flight_data,x='destination_city',hue='class',palette=col)
sns.move_legend(bx,"right")
plt.title('No.of flights per destination city')
plt.xlabel('Destination Cities')
plt.ylabel('No.of flights')
for c in bx.containers:
  bx.bar_label(c)
plt.show()
```



From both charts,

- Economy class:- Delhi has the highest number, and
- Business class:- Mumbai is the city with highest no.of flights

e. Statistical info of the dataset

```
flight_data.describe()
```

	duration	days_left	price
count	300153.000000	300153.000000	300153.000000
mean	12.221021	26.004751	20889.660523
std	7.191997	13.561004	22697.767366
min	0.830000	1.000000	1105.000000
25%	6.830000	15.000000	4783.000000
50%	11.250000	26.000000	7425.000000
75%	16.170000	38.000000	42521.000000
max	49.830000	49.000000	123071.000000

f. Viewing ticket price by each airline and class

```
flight_data[['airline','price','class']].sort_values(by='price',ascending=False)
```

	airline	price	class
261377	Vistara	123071	Business
216096	Vistara	117307	Business
215859	Vistara	116562	Business
277345	Vistara	115211	Business
270999	Vistara	114705	Business
...	...	...	...
204375	AirAsia	1105	Economy
204376	GO_FIRST	1105	Economy
206598	Indigo	1105	Economy
206599	Indigo	1105	Economy
205024	Indigo	1105	Economy

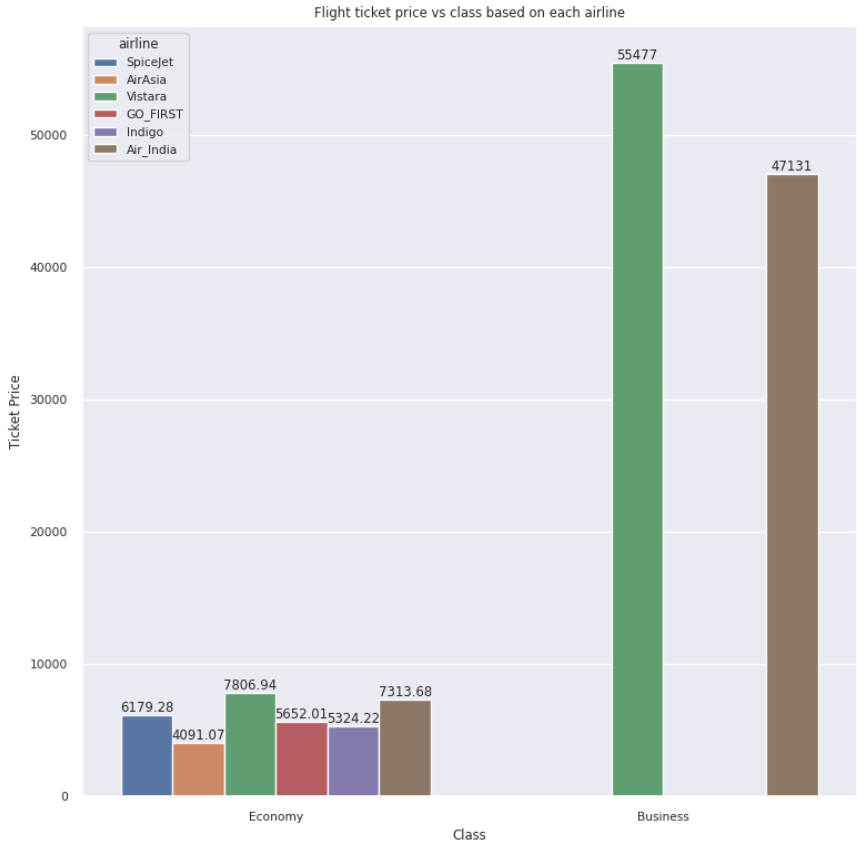
300153 rows × 3 columns

Among the various airlines, Vistara charges highest price under the business class.

g. Ticket price vs class based on different airlines

```
sns.set(font_scale=0.7)
plt.figure(figsize=(9,9))
x=sns.barplot(data=flight_data,x='class',y='price',hue='airline',errorbar=None)
for i in x.containers:
  x.bar_label(i)
plt.xlabel('Class')
plt.ylabel('Ticket Price')
plt.title('Flight ticket price vs class based on each airline')
```

```
Text(0.5, 1.0, 'Flight ticket price vs class based on each airline')
```



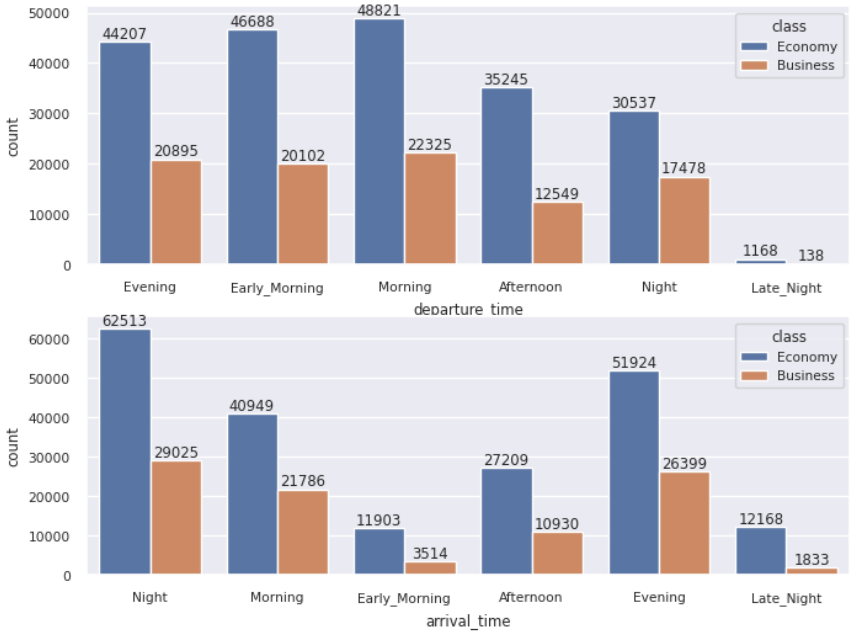
The ticket price charged by Vistara is the highest under both classes, and AirAsia offers the lowest under Economy class.

h. Plotting No.of flights per class under different departure and arrival time.

```
sns.set(font_scale=0.7)
plt.figure(figsize=(8,6))

plt.subplot(2,1,1)
cl=sns.countplot(data=flight_data,x='departure_time',hue='class')
for l in cl.containers:
  cl.bar_label(l)

plt.subplot(2,1,2)
cl=sns.countplot(data=flight_data,x='arrival_time',hue='class')
for l in cl.containers:
  cl.bar_label(l)
```



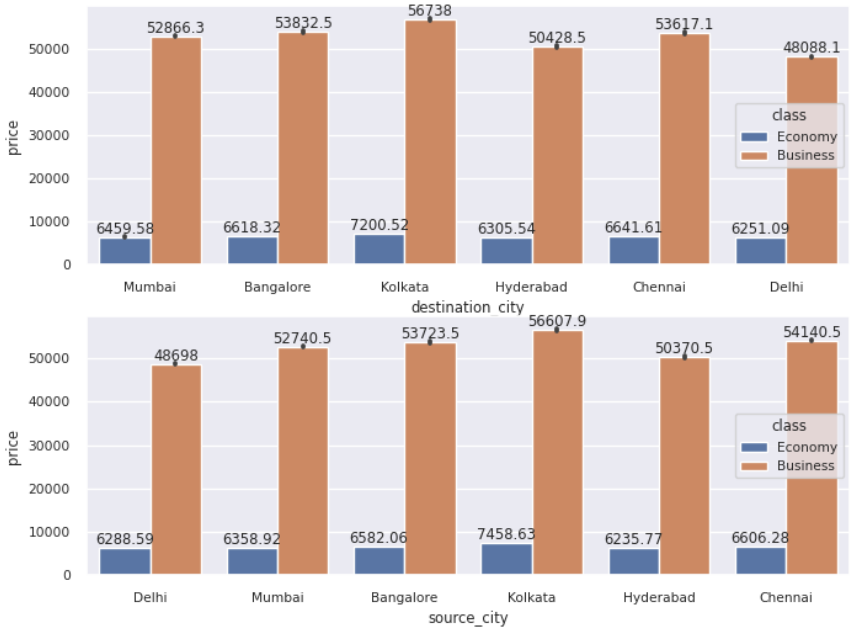
This graph shows that, more morning flights are departed as well as more night flights arrive at the airport.

i. Analysing ticket price vs destination and source cities base on each class

```
sns.set(font_scale=0.7)
plt.figure(figsize=(8,6))

plt.subplot(2,1,1)
cl=sns.barplot(data=flight_data,x='destination_city',y='price',hue='class')
for l in cl.containers:
  cl.bar_label(l)

plt.subplot(2,1,2)
cl=sns.barplot(data=flight_data,x='source_city',y='price',hue='class')
for l in cl.containers:
  cl.bar_label(l)
```



Kolkata's flight is the costliest

j. Analysing duration of flights

```
flight_data['duration'].describe()
```

```
count    300153.000000
mean         12.221021
std           7.191997
min           0.830000
25%           6.830000
50%          11.250000
75%          16.170000
max          49.830000
Name: duration, dtype: float64
```

(4) Feature Engineering

1. Checking for outliers in price column

```
sns.boxplot(data=flight_data,x='price')
```

```
<Axes: xlabel='price'>
```