Flight Passengers Analysis

This is a Python Project of data analysis, where we take a dataset of a flight, by which we have to find some insights and solutions of the problem that occur in the flight. To explains Every point line by line I am going to visualize the data by which i gonna get some insights from it.

Let's Get Started.....

First we have to load the dataset on which we are going to perform that action, the data is available on Kaggle so I am going to fetch it from there.

Link of the dataset : https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction

Data Cleaning

In this analysis we have to find the reason, by which the passengers take only one flight more as compare to another.

So we have to find the solution for that problem and also we have to find some insights for that dataset and also we have to perform some EDA task and for that we have to clean the dataset to do it.

Now first work is to perform some cleaning task for dataset.

Importing the python modules for performing the different task on dataset.

```
import pandas as pd

import matplotlib.pyplot as plt
```

After loading the modules of python, I am going to load the dataset to perform the EDA Task means Exploratory Data

Analysis, but for EDA first we have to clean the dataset.

Loading the data set.

```
flight_df = pd.read_csv("data//Clean_Dataset.csv")
```

Now first we check the top 5 values of dataset

```
flight_df.head()
```

According to head method we get the top 5 data of dataset, now first I need to analyse the data and explain about

all the columns that are present in the given dataset.

Let me find the columns name separately

```
print(flight_df.columns)
```

We have total 12 columns and some of them are not important so we have to remove some column from it.

Now we have to find the summary of the dataset.

```
print(flight_df.info())
```

According to the info method, I analyze that their is no null values in dataset, and all the columns are of object type except days_left and price.

In the dataset, duration, days_left, and price are not important column, so I am going to remove it.

```
print(flight_df.drop(["duration", "days_left", "price"], axis=1))
```

Next I am describing the dataset for the statical analysis

```
print(flight_df.describe(include="all"))
```

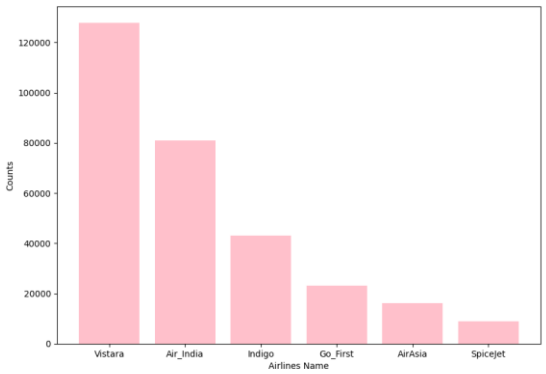
Next I am going to find the category and its value count from airline column

```
print(flight_df["airline"].value_counts())
```

According to the value_count method, I get that Vistara airlines are used so much by peoples, and spiceJet airline is used very less.

For Making this analysis more easy, I am going to plot a bar graph to represent these flights in visual form.

Visual representation of Count of airlines.



*2 - **plt.figure(figsize=(10, 7))***

flight_name = ["Vistara", "Air_India", "Indigo", "Go_First", "AirAsia", "SpiceJet"]

flight_count = flight_df["airline"].value_counts()

plt.bar(flight_name, flight_count, color="pink")

plt.xlabel("Airlines Name")

plt.ylabel("Counts")

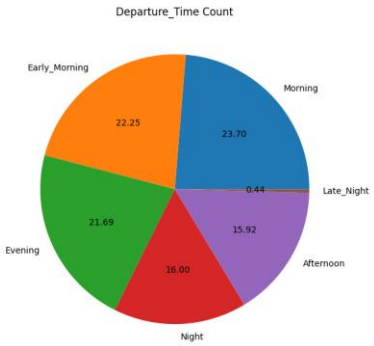
plt.show()

Now we have to count the total counts of the city form where the flight is board.

print(flight_df["departure_time"].value_counts())

According to count function, I analyze that maximum passengers are taking their flight in Morning, and very less passengers taking their flight in late night.

Now to make the analysis more easy , i am going to plat a bar chart to represent the departure time.



*3 - **plt.figure(figsize=(10, 7))***

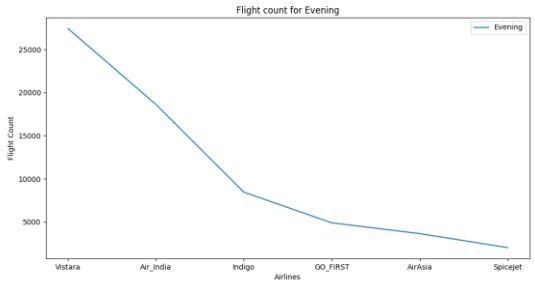
departure_count = flight_df["departure_time"].value_counts()

plt.title('Departure_Time Count')

plt.pie(departure_count, autopct='%.2f', labels=departure_count.index)

plt.show()

Now I am going to find the flight count value according to departure time.



4 - Evening departure flight Count

evening = flight_df[flight_df["departure_time"] == "Evening"]

print(evening["airline"].value_counts())

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

plt.title("Flight count for Evening")

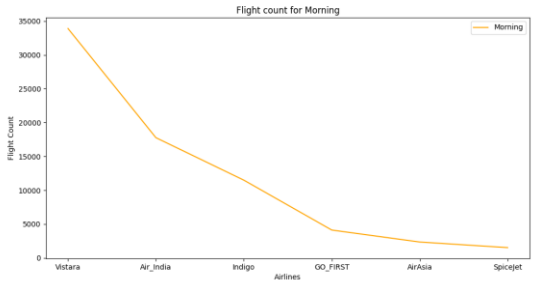
plt.plot(evening["airline"].value_counts(), label="Evening")

plt.xlabel("Airlines")

plt.ylabel("Flight Count")

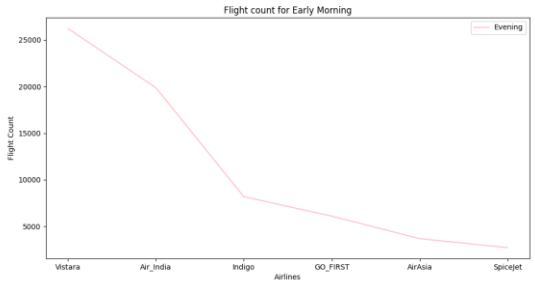
plt.legend()

plt.show()



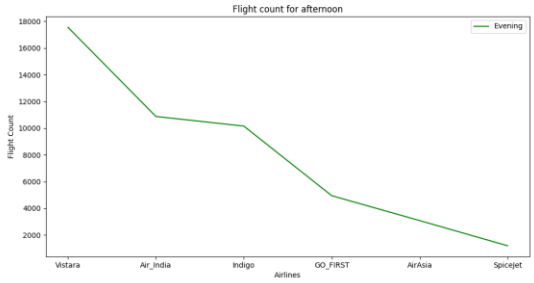
*5 - **Morning departure Flight Count***

1. *morning = flight_df[flight_df["departure_time"] == "Morning"]*
2. *print(morning["airline"].value_counts())*
3. *plt.figure(figsize=(12, 6))*
4. *plt.title("Flight count for Morning")*
5. *plt.plot(morning["airline"].value_counts(), label="Morning", color="orange")*
6. *plt.xlabel("Airlines")*
7. *plt.ylabel("Flight Count")*
8. *plt.legend()*
9. *plt.show()*



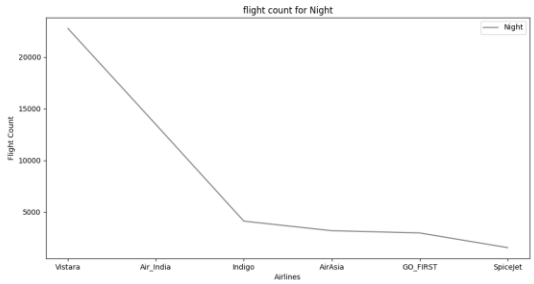
*6 - **Early Morning departure flight count***

1. *early_morning = flight_df[flight_df["departure_time"] == "Early_Morning"]*
2. *print(early_morning["airline"].value_counts())*
3. *plt.figure(figsize=(12, 6))*
4. *plt.title("Flight count for Early Morning")*
5. *plt.plot(early_morning["airline"].value_counts(), color="pink", label="Evening")*
6. *plt.xlabel("Airlines")*
7. *plt.ylabel("Flight Count")*
8. *plt.legend()*
9. *plt.show()*



*7 - **Afternoon departure flight count.***

1. *afternoon = flight_df[flight_df["departure_time"] == "Afternoon"]*
2. *print(afternoon["airline"].value_counts())*
3. *plt.figure(figsize=(12, 6))*
4. *plt.title("Flight count for afternoon")*
5. *plt.plot(afternoon["airline"].value_counts(), color="green", label="Evening")*
6. *plt.xlabel("Airlines")*
7. *plt.ylabel("Flight Count")*
8. *plt.legend()*
9. *plt.show()*



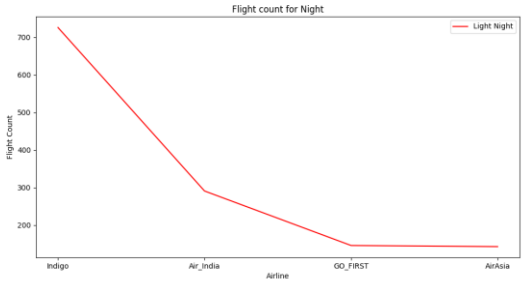
*8 - **Night departure flight count***

1. *night = flight_df[flight_df["departure_time"] == "Night"]*
2. *print(night["airline"].value_counts())*
3. *plt.figure(figsize=(12, 6))*
4. *plt.title("flight count for Night")*
5. *plt.plot(night["airline"].value_counts(), color="gray", label="Night")*
6. *plt.xlabel("Airlines")*

7. plt.ylabel("Flight Count")

8. plt.legend()

9. plt.show()



9 - ***Late Night departure flight count***

1. late_night = flight_df[flight_df["departure_time"] == "Late_Night"]

2. print(late_night["airline"].value_counts())

3. plt.figure(figsize=(12, 6))

4. plt.title("Flight count for Night")

5. plt.plot(late_night["airline"].value_counts(), color="red", label="Light Night")

6. plt.xlabel("Airline")

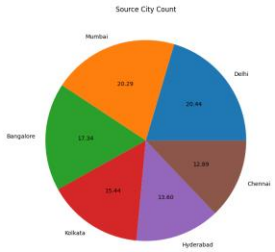
7. plt.ylabel("Flight Count")

8. plt.legend()

9. plt.show()

From the above analysis and visualization Vistara is the only flight who board by passengers.

Source city Analysis



10 -

1. print(flight_df["source_city"].value_counts())

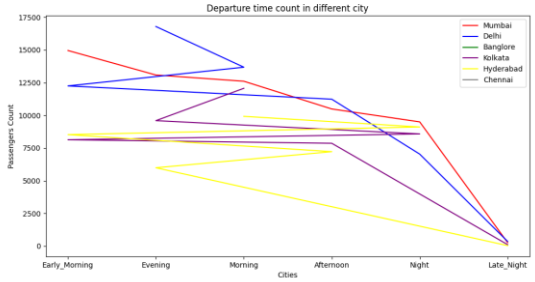
According to above analysis Delhi is the source city, from where the maximum passengers board their flight, and Chennai is the source city from where the passengers board less flight.

Now I just create a visual presentation for better presentation.

1. *source_city_count = flight_df["source_city"].value_counts()*
2. *print(source_city_count)*
3. *plt.figure(figsize=(10, 7))*
4. *plt.title('Source City Count')*
5. *plt.pie(source_city_count, autopct='%.2f', labels=source_city_count.index)*
6. *plt.show()*

According to this analysis, I know that in delhi city maximum people use the flight.

Checking the departure time in different city.



11 -

1. *mumbai = flight_df[flight_df["source_city"] == "Mumbai"]*
2. *delhi = flight_df[flight_df["source_city"] == "Delhi"]*
3. *banglore = flight_df[flight_df["source_city"] == "Banglore"]*
4. *kolkata = flight_df[flight_df["source_city"] == "Kolkata"]*
5. *hyderabad = flight_df[flight_df["source_city"] == "Hyderabad"]*
6. *chennai = flight_df[flight_df["source_city"] == "Chennnai"]*
7. *plt.figure(figsize=(12, 6))*
8. *plt.title("Departure time count in different city")*
9. *plt.plot(mumbai["departure_time"].value_counts(), color="red", label="Mumbai")*
10. *plt.plot(delhi["departure_time"].value_counts(), color="blue", label="Delhi")*
11. *plt.plot(banglore["departure_time"].value_counts(), color="green", label="Banglore")*

12. *plt.plot(kolkata["departure_time"].value_counts(), color="purple", label="Kolkata")*

13. *plt.plot(hyderabad["departure_time"].value_counts(), color="yellow", label="Hyderabad")*

14. *plt.plot(chennai["departure_time"].value_counts(), color="grey", label="Chennai")*

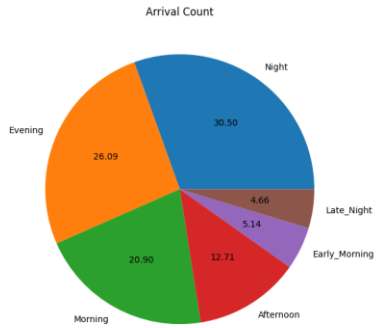
15. *plt.xlabel("Cities")*

16. *plt.ylabel("Passengers Count")*

17. *plt.legend()*

18. *plt.show()*

Now we have to count the total counts of arrival time form where the flight is land.



12 - print(flight_df["arrival_time"].value_counts())

Now to make the analysis more easy , I am going to plat a bar chart to represent the arrival time.

1. *plt.figure(figsize=(10, 7))*

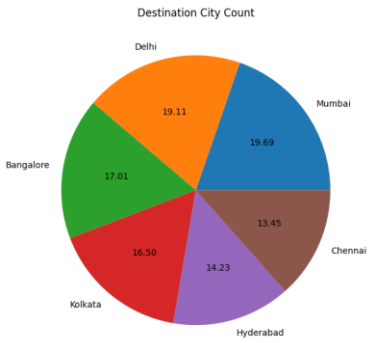
2. *arrival_count = flight_df["arrival_time"].value_counts()*

3. *plt.title('Arrival Count')*

4. *plt.pie(arrival_count, autopct='%.2f', labels=arrival_count.index)*

5. *plt.show()*

Destination City Count



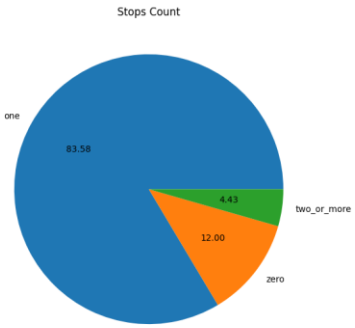
13 -

1. *print(flight_df["destination_city"].value_counts())*

Now I just create a visual presentation for better presentation.

1. *destination_count = flight_df["destination_city"].value_counts()*
2. *print(destination_count)*
3. *plt.figure(figsize=(10, 7))*
4. *plt.title('Destination City Count')*
5. *plt.pie(destination_count, autopct='%.2f', labels=destination_count.index)*
6. *plt.show()*

Maximum stops city count



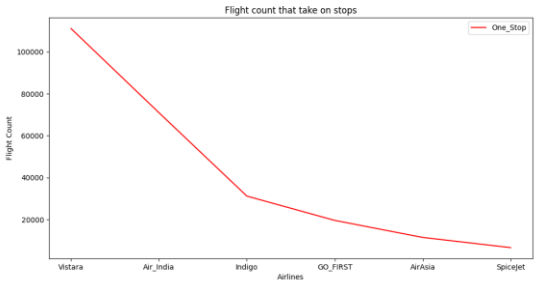
14 -

1. *print(flight_df["stops"].value_counts())*

Visual representation of above data

```
1. stops_count = flight_df["stops"].value_counts()
2. print(stops_count)
3. plt.figure(figsize=(10, 7))
4. plt.title('Stops Count')
5. plt.pie(stops_count, autopct='%.2f', labels=stops_count.index)
6. plt.show()
```

Analysis said that count of one stop is maximum as compare to others.

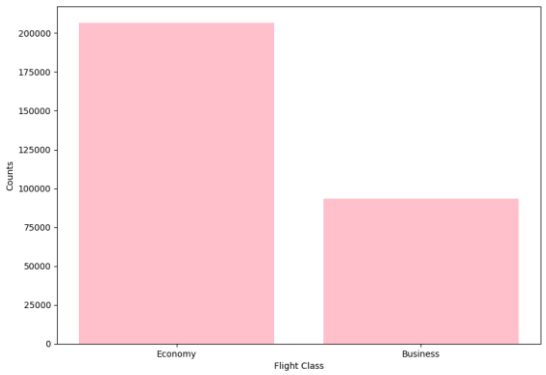


15 - ***Flight_count that takes one stop***

```
1. one = flight_df[flight_df["stops"] == "one"]
2. print(one["airline"].value_counts())
3. plt.figure(figsize=(12, 6))
4. plt.title("Flight count that take on stops")
5. plt.plot(one["airline"].value_counts(), label="One_Stop", color="red")
6. plt.xlabel("Airlines")
7. plt.ylabel("Flight Count")
8. plt.legend()
9. plt.show()
```

In One stop Vistara is the top flight.

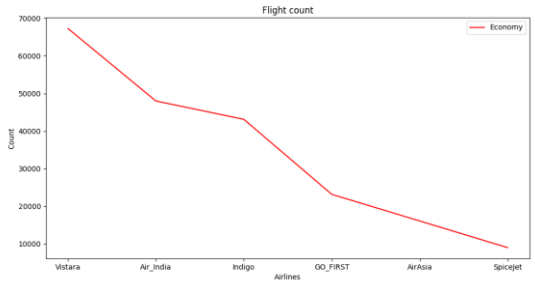
Flight classes



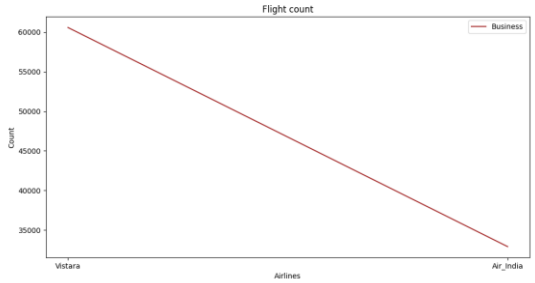
*16 - **Counting of passengers class***

```
1. print(flight_df["class"].value_counts())
2. plt.figure(figsize=(10, 7))
3. flight_name = ["Economy", "Business"]
4. flight_count = flight_df["class"].value_counts()
5. plt.bar(flight_name, flight_count, color="pink")
6. plt.xlabel("Flight Class")
7. plt.ylabel("Counts")
8. plt.show()
```

Maximum Passengers who board their flight are taking economy class.



Maximum Passengers who board their flight are taking Business class.



*17 - **Flight Count for max booking in Business***

```
1. business = flight_df[flight_df["class"] == "Business"]
2. print(business["airline"].value_counts())
3. plt.figure(figsize=(12, 6))
4. plt.title("Flight count")
5. plt.plot(business["airline"].value_counts(), label="Business", color="brown")
6. plt.xlabel("Airlines")
7. plt.ylabel("Count")
8. plt.legend()
9. plt.show()
```

Vistara is the airline in which passengers book their flight in business class.

Insights

- **Vistara is the only flight who is comes in top in every analysis.**
- **In this flight analysis we see that Vistara do something good by which he is on the top, So we have to improve the other flights by adopting some changes from Vistara.**
- **For improving the other flight first we have to decrease the prices of morning flight, by this we overcome from two thing one is pasangers attract to the flight and also the average passengers prefer morning time so no. of passengers.**
- **Also we have to manage the departure time, means scheduling of flights.**