INTRODUCTION

- This data is given by an airline organization.Original name of the company is not given so we call it as Invistico_Airline.
- The main objective of the analysis is to predict whether a potential customer will be satisfied with the service offered and to identify which factors expressed by customers most influence satisfaction.

The selected dataset contains a total of 23 variables and 129,880 observations. The variables are:

1. satisfaction: The overall satisfaction level of the customer. It is a categorical variable with options "satisfied" or "dissatisfied".
2. Gender: The gender of the customer. It is a categorical variable with options "male" or "female".
3. Customer Type: Whether the customer is a "loyal customer" or a "disloyal customer".
4. Age: The age of the customer.
5. Type of Travel: This column indicates the purpose of the customer's travel. It is a catagorical variable with two possible values: "Personal Travel" or "Business travel".
6. Class: The class of travel, such as "Eco","Eco Plus" or "Business"
7. Flight Distance: The distance of the flight.
8. Seat comfort: Customer rating of seat comfort.
9. Departure/Arrival time convenient: Customer rating of convenience of departure/arrival times.
10. Food and drink: Customer rating of food and drink quality.
11. Gate location: Customer rating of gate location.
12. Inflight wifi service: Customer rating of inflight Wi-Fi service.
13. Inflight entertainment: Customer rating of inflight entertainment options.
14. Online support: Customer rating of online customer support.
15. Ease of Online booking: Customer rating of ease of online booking.
16. On-board service: Customer rating of on-board service provided by the airline.
17. Leg room service: Customer rating of leg room service provided during the flight.
18. Baggage handling: Customer rating of baggage handling.
19. Checkin service: Customer rating of check-in service.
20. Cleanliness: Customer rating of cabin cleanliness.
21. Online boarding: Customer rating of online boarding process.
22. Departure Delay in Minutes: The departure delay in minutes for each flight.
23. Arrival Delay in Minutes: The arrival delay in minutes for each flight

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

```
df=pd.read_csv("Invistico_Airline.csv")

df.head()

  satisfaction  Gender   Customer Type  Age   Type of Travel     Class
\
0    satisfied  Female  Loyal Customer   65  Personal Travel       Eco

1    satisfied    Male  Loyal Customer   47  Personal Travel  Business

2    satisfied  Female  Loyal Customer   15  Personal Travel       Eco

3    satisfied  Female  Loyal Customer   60  Personal Travel       Eco

4    satisfied  Female  Loyal Customer   70  Personal Travel       Eco


   Flight Distance  Seat comfort  Departure/Arrival time convenient  \
0              265             0                                  0
1             2464             0                                  0
2             2138             0                                  0
3              623             0                                  0
4              354             0                                  0

   Food and drink  ...  Online support  Ease of Online booking  \
0               0  ...               2                       3
1               0  ...               2                       3
2               0  ...               2                       2
3               0  ...               3                       1
4               0  ...               4                       2

   On-board service  Leg room service  Baggage handling  Checkin
service  \
0                 3                 0                 3
5
1                 4                 4                 4
2
2                 3                 3                 4
4
3                 1                 0                 1
4
4                 2                 0                 2
4

   Cleanliness  Online boarding  Departure Delay in Minutes  \
0            3                2                           0
1            3                2                         310
2            4                2                           0
3            1                3                           0
4            2                5                           0
```

```
   Arrival Delay in Minutes
0                       0.0
1                     305.0
2                       0.0
3                       0.0
4                       0.0

[5 rows x 23 columns]
```

```
#Column information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 23 columns):
 #   Column                             Non-Null Count   Dtype  
---  ------                             --------------   -----  
 0   satisfaction                       129880 non-null  object 
 1   Gender                             129880 non-null  object 
 2   Customer Type                      129880 non-null  object 
 3   Age                                129880 non-null  int64  
 4   Type of Travel                     129880 non-null  object 
 5   Class                              129880 non-null  object 
 6   Flight Distance                    129880 non-null  int64  
 7   Seat comfort                       129880 non-null  int64  
 8   Departure/Arrival time convenient  129880 non-null  int64  
 9   Food and drink                     129880 non-null  int64  
 10  Gate location                      129880 non-null  int64  
 11  Inflight wifi service              129880 non-null  int64  
 12  Inflight entertainment             129880 non-null  int64  
 13  Online support                     129880 non-null  int64  
 14  Ease of Online booking             129880 non-null  int64  
 15  On-board service                   129880 non-null  int64  
 16  Leg room service                   129880 non-null  int64  
 17  Baggage handling                   129880 non-null  int64  
 18  Checkin service                    129880 non-null  int64  
 19  Cleanliness                        129880 non-null  int64  
 20  Online boarding                    129880 non-null  int64  
 21  Departure Delay in Minutes         129880 non-null  int64  
 22  Arrival Delay in Minutes           129487 non-null  float64
dtypes: float64(1), int64(17), object(5)
memory usage: 22.8+ MB
```

```
df.size
```

```
2987240
```

```
df.shape
```

```
(129880, 23)
```

```python
#datatypes of our columns
df.dtypes
```

```
satisfaction                          object
Gender                                object
Customer Type                         object
Age                                    int64
Type of Travel                        object
Class                                 object
Flight Distance                        int64
Seat comfort                           int64
Departure/Arrival time convenient      int64
Food and drink                         int64
Gate location                          int64
Inflight wifi service                  int64
Inflight entertainment                 int64
Online support                         int64
Ease of Online booking                 int64
On-board service                       int64
Leg room service                       int64
Baggage handling                       int64
Checkin service                        int64
Cleanliness                            int64
Online boarding                        int64
Departure Delay in Minutes             int64
Arrival Delay in Minutes             float64
dtype: object
```

```python
#Now we can check the statistical information of the data
df.describe(include='all')
```

```
       satisfaction  Gender   Customer Type            Age   Type of
Travel  \
count        129880  129880          129880  129880.000000
129880
unique            2       2               2            NaN
2
top       satisfied  Female  Loyal Customer            NaN  Business
travel
freq          71087   65899          106100            NaN
89693
mean            NaN     NaN             NaN      39.427957
NaN
std             NaN     NaN             NaN      15.119360
NaN
min             NaN     NaN             NaN       7.000000
NaN
25%             NaN     NaN             NaN      27.000000
NaN
50%             NaN     NaN             NaN      40.000000
```

```
NaN
75%              NaN     NaN              NaN      51.000000
NaN
max              NaN     NaN              NaN      85.000000
NaN

           Class  Flight Distance   Seat comfort  \
count     129880    129880.000000  129880.000000
unique         3              NaN            NaN
top     Business              NaN            NaN
freq       62160              NaN            NaN
mean         NaN      1981.409055       2.838597
std          NaN      1027.115606       1.392983
min          NaN        50.000000       0.000000
25%          NaN      1359.000000       2.000000
50%          NaN      1925.000000       3.000000
75%          NaN      2544.000000       4.000000
max          NaN      6951.000000       5.000000

        Departure/Arrival time convenient  Food and drink  ...  \
count                       129880.000000   129880.000000  ...
unique                                NaN             NaN  ...
top                                   NaN             NaN  ...
freq                                  NaN             NaN  ...
mean                             2.990645        2.851994  ...
std                              1.527224        1.443729  ...
min                              0.000000        0.000000  ...
25%                              2.000000        2.000000  ...
50%                              3.000000        3.000000  ...
75%                              4.000000        4.000000  ...
max                              5.000000        5.000000  ...

        Online support  Ease of Online booking  On-board service  \
count    129880.000000           129880.000000     129880.000000
unique             NaN                     NaN               NaN
top                NaN                     NaN               NaN
freq               NaN                     NaN               NaN
mean          3.519703                3.472105          3.465075
std           1.306511                1.305560          1.270836
min           0.000000                0.000000          0.000000
25%           3.000000                2.000000          3.000000
50%           4.000000                4.000000          4.000000
75%           5.000000                5.000000          4.000000
max           5.000000                5.000000          5.000000

        Leg room service  Baggage handling  Checkin service
Cleanliness  \
count      129880.000000     129880.000000    129880.000000
129880.000000
unique               NaN               NaN              NaN
```

```
NaN
top                    NaN                  NaN                NaN
NaN
freq                   NaN                  NaN                NaN
NaN
mean              3.485902             3.695673           3.340807
3.705759
std               1.292226             1.156483           1.260582
1.151774
min               0.000000             1.000000           0.000000
0.000000
25%               2.000000             3.000000           3.000000
3.000000
50%               4.000000             4.000000           3.000000
4.000000
75%               5.000000             5.000000           4.000000
5.000000
max               5.000000             5.000000           5.000000
5.000000

        Online boarding  Departure Delay in Minutes  Arrival Delay in
Minutes
count     129880.000000               129880.000000
129487.000000
unique              NaN                         NaN
NaN
top                 NaN                         NaN
NaN
freq                NaN                         NaN
NaN
mean           3.352587                   14.713713
15.091129
std            1.298715                   38.071126
38.465650
min            0.000000                    0.000000
0.000000
25%            2.000000                    0.000000
0.000000
50%            4.000000                    0.000000
0.000000
75%            4.000000                   12.000000
13.000000
max            5.000000                 1592.000000
1584.000000

[11 rows x 23 columns]

#checking whether dataset is having any duplicate values
df.duplicated().sum()
```

```
0

#Checking whether the dataset is having any missing values
df.isnull().sum()
```

```
satisfaction                           0
Gender                                 0
Customer Type                          0
Age                                    0
Type of Travel                         0
Class                                  0
Flight Distance                        0
Seat comfort                           0
Departure/Arrival time convenient      0
Food and drink                         0
Gate location                          0
Inflight wifi service                  0
Inflight entertainment                 0
Online support                         0
Ease of Online booking                 0
On-board service                       0
Leg room service                       0
Baggage handling                       0
Checkin service                        0
Cleanliness                            0
Online boarding                        0
Departure Delay in Minutes             0
Arrival Delay in Minutes             393
dtype: int64
```

```
#In the above data the feature "Arrival Delay in Minutes" have some
null value so next we remove the null values.
#we are removing null values using fillna.

df['Arrival Delay in Minutes'] = df['Arrival Delay in
Minutes'].fillna(df['Arrival Delay in Minutes'].mean())

df.isnull().sum()
```

```
satisfaction                         0
Gender                               0
Customer Type                        0
Age                                  0
Type of Travel                       0
Class                                0
Flight Distance                      0
Seat comfort                         0
Departure/Arrival time convenient    0
Food and drink                       0
Gate location                        0
```

```
Inflight wifi service                0
Inflight entertainment               0
Online support                       0
Ease of Online booking               0
On-board service                     0
Leg room service                     0
Baggage handling                     0
Checkin service                      0
Cleanliness                          0
Online boarding                      0
Departure Delay in Minutes           0
Arrival Delay in Minutes             0
dtype: int64
```

```
#checking unique values
features=['satisfaction','Gender','Customer Type','Class','Type of Travel']
for i in features:
    print(df[i].unique(),i)
```

```
['satisfied' 'dissatisfied'] satisfaction
['Female' 'Male'] Gender
['Loyal Customer' 'disloyal Customer'] Customer Type
['Eco' 'Business' 'Eco Plus'] Class
['Personal Travel' 'Business travel'] Type of Travel
```

```
#Distribution of categorical variables
categorical_columns=df.select_dtypes(include=['object'])
for col in categorical_columns:
    print(df[col].value_counts())
```

```
satisfaction
satisfied       71087
dissatisfied    58793
Name: count, dtype: int64
Gender
Female    65899
Male      63981
Name: count, dtype: int64
Customer Type
Loyal Customer       106100
disloyal Customer     23780
Name: count, dtype: int64
Type of Travel
Business travel    89693
Personal Travel    40187
Name: count, dtype: int64
Class
Business    62160
Eco         58309
```

```
Eco Plus     9411
Name: count, dtype: int64
```

```python
#Distribution of numerical variables
numerical_columns=df.select_dtypes(include=['int','float'])
for col in numerical_columns:
    print(df[col].value_counts())
```

```
Age
39    3692
25    3511
40    3209
44    3104
41    3089
      ... 
74      61
76      60
79      52
78      44
85      25
Name: count, Length: 75, dtype: int64
Flight Distance
1963    92
1812    88
1639    87
1981    86
1789    86
        ..
4222     1
5049     1
5378     1
5613     1
4260     1
Name: count, Length: 5398, dtype: int64
Seat comfort
3    29183
2    28726
4    28398
1    20949
5    17827
0     4797
Name: count, dtype: int64
Departure/Arrival time convenient
4    29593
5    26817
3    23184
2    22794
1    20828
0     6664
Name: count, dtype: int64
```

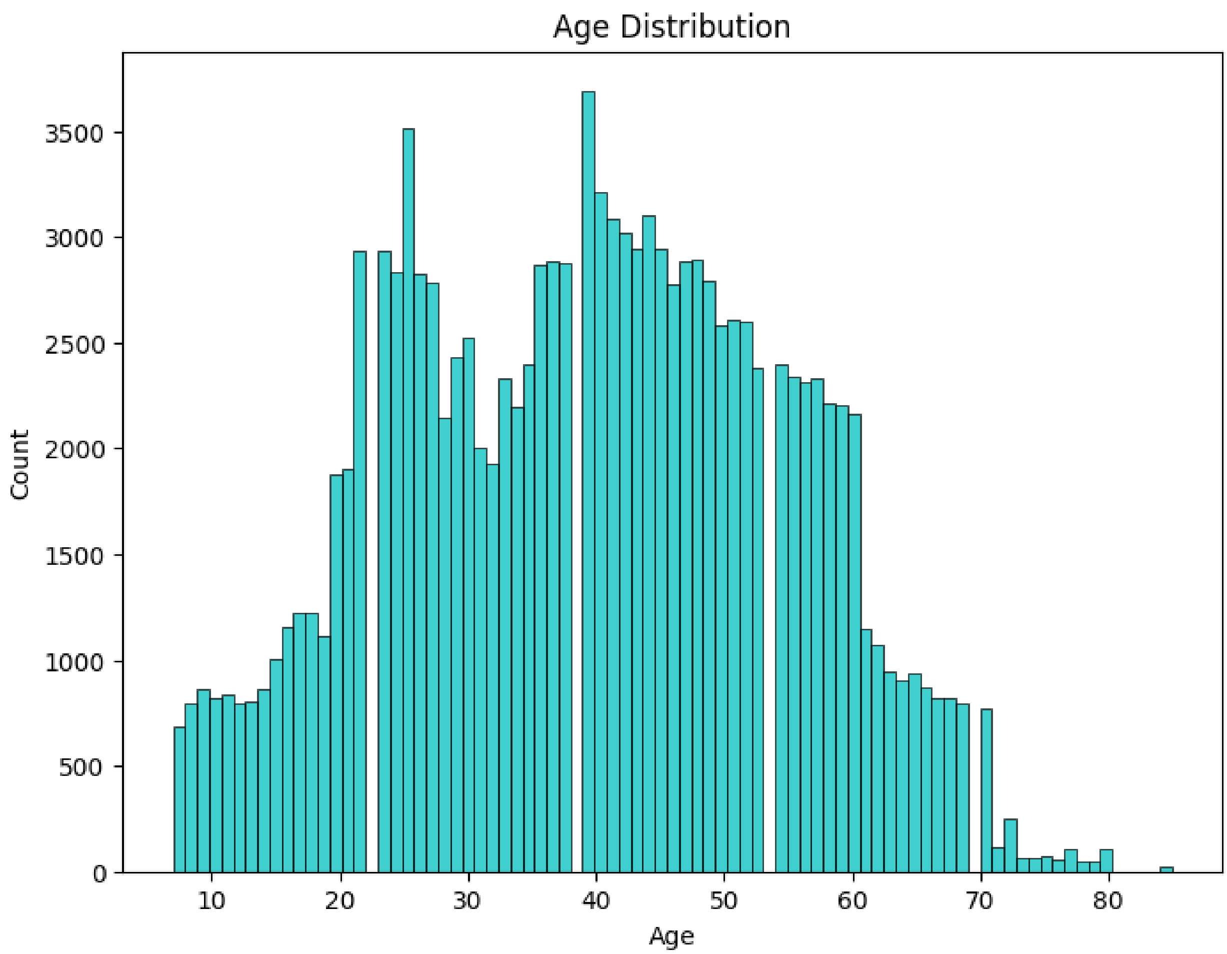
```
Food and drink
3    28150
4    27216
2    27146
1    21076
5    20347
0     5945
Name: count, dtype: int64
Gate location
3    33546
4    30088
2    24518
1    22565
5    19161
0        2
Name: count, dtype: int64
Inflight wifi service
4    31560
5    28830
3    27602
2    27045
1    14711
0      132
Name: count, dtype: int64
Inflight entertainment
4    41879
5    29831
3    24200
2    19183
1    11809
0     2978
Name: count, dtype: int64
Online support
4    41510
5    35563
3    21609
2    17260
1    13937
0        1
Name: count, dtype: int64
Ease of Online booking
4    39920
5    34137
3    22418
2    19951
1    13436
0       18
Name: count, dtype: int64
On-board service
```

```
4    40675
5    31724
3    27037
2    17174
1    13265
0        5
Name: count, dtype: int64
Leg room service
4    39698
5    34385
3    22467
2    21745
1    11141
0      444
Name: count, dtype: int64
Baggage handling
4    48240
5    35748
3    24485
2    13432
1     7975
Name: count, dtype: int64
Checkin service
4    36481
3    35538
5    27005
2    15486
1    15369
0        1
Name: count, dtype: int64
Cleanliness
4    48795
5    35916
3    23984
2    13412
1     7768
0        5
Name: count, dtype: int64
Online boarding
4    35181
3    30780
5    29973
2    18573
1    15359
0       14
Name: count, dtype: int64
Departure Delay in Minutes
0      73356
1       3682
```

```
2        2855
3        2535
4        2309
       ...
366        1
569        1
419        1
411        1
320        1
Name: count, Length: 466, dtype: int64
Arrival Delay in Minutes
0.0      72753
1.0       2747
2.0       2587
3.0       2442
4.0       2373
         ...
443.0        1
418.0        1
608.0        1
429.0        1
500.0        1
Name: count, Length: 473, dtype: int64
```

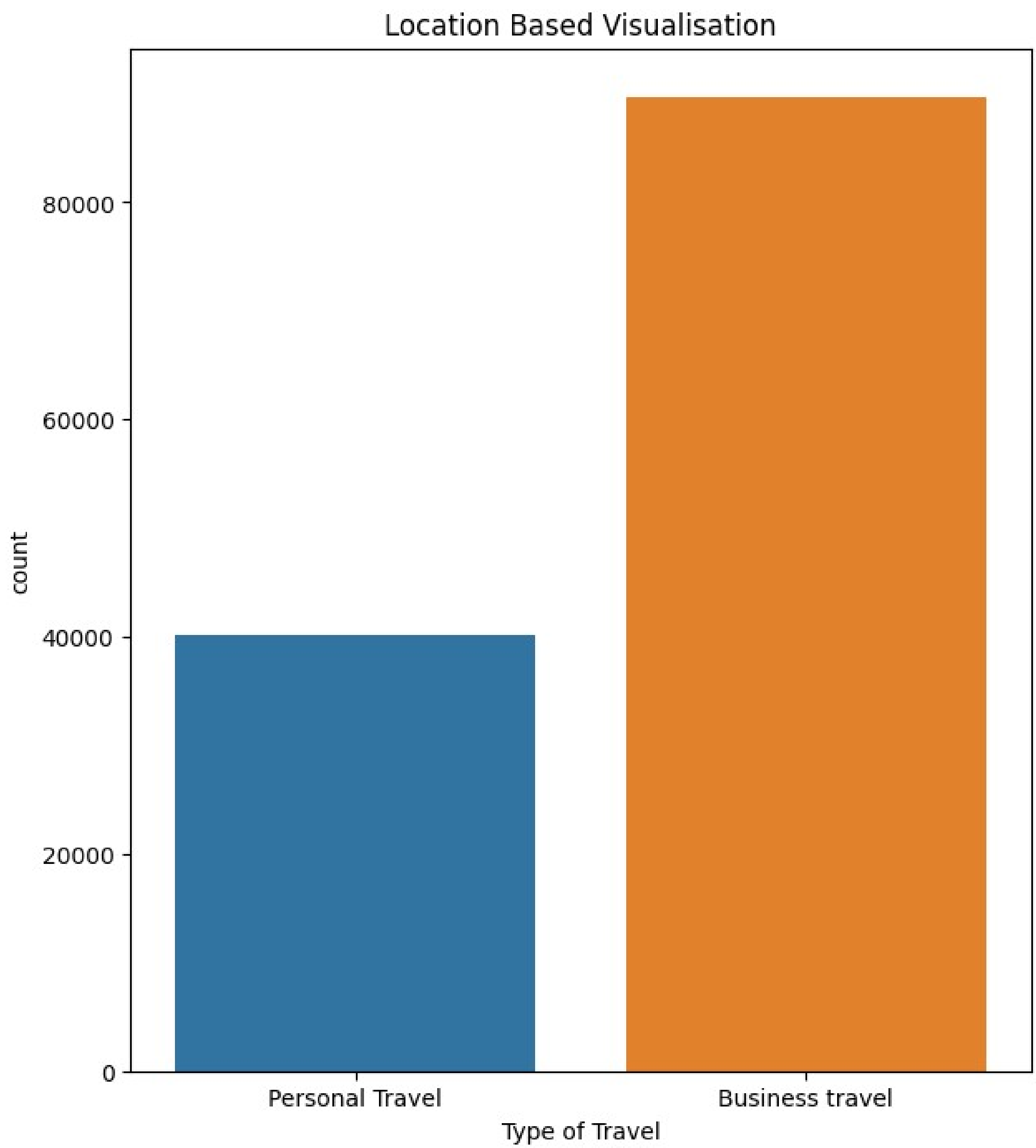
DATA VISUALIZATION

```
#Histoplot
# Distribution of Age
plt.figure(figsize=(8, 6))
sns.histplot(data=df, x='Age',color='c')
plt.title('Age Distribution')
plt.show()
```



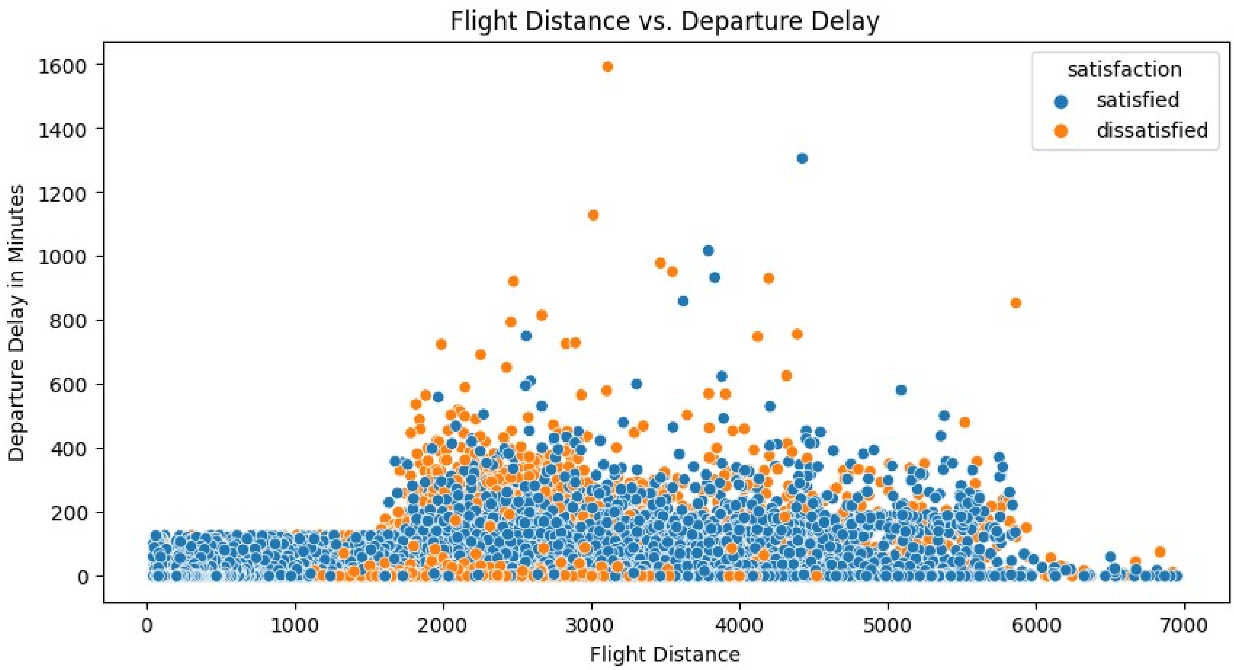
From the Age Distribution It is clear that most of the people in the dataset have the age 39 and 26.

```
#Countplot
plt.figure(figsize=(7,8))
sns.countplot(x='Type of Travel',data=df)
plt.title("Type of Travel")
plt.show()
```



Most of them are Business Travels

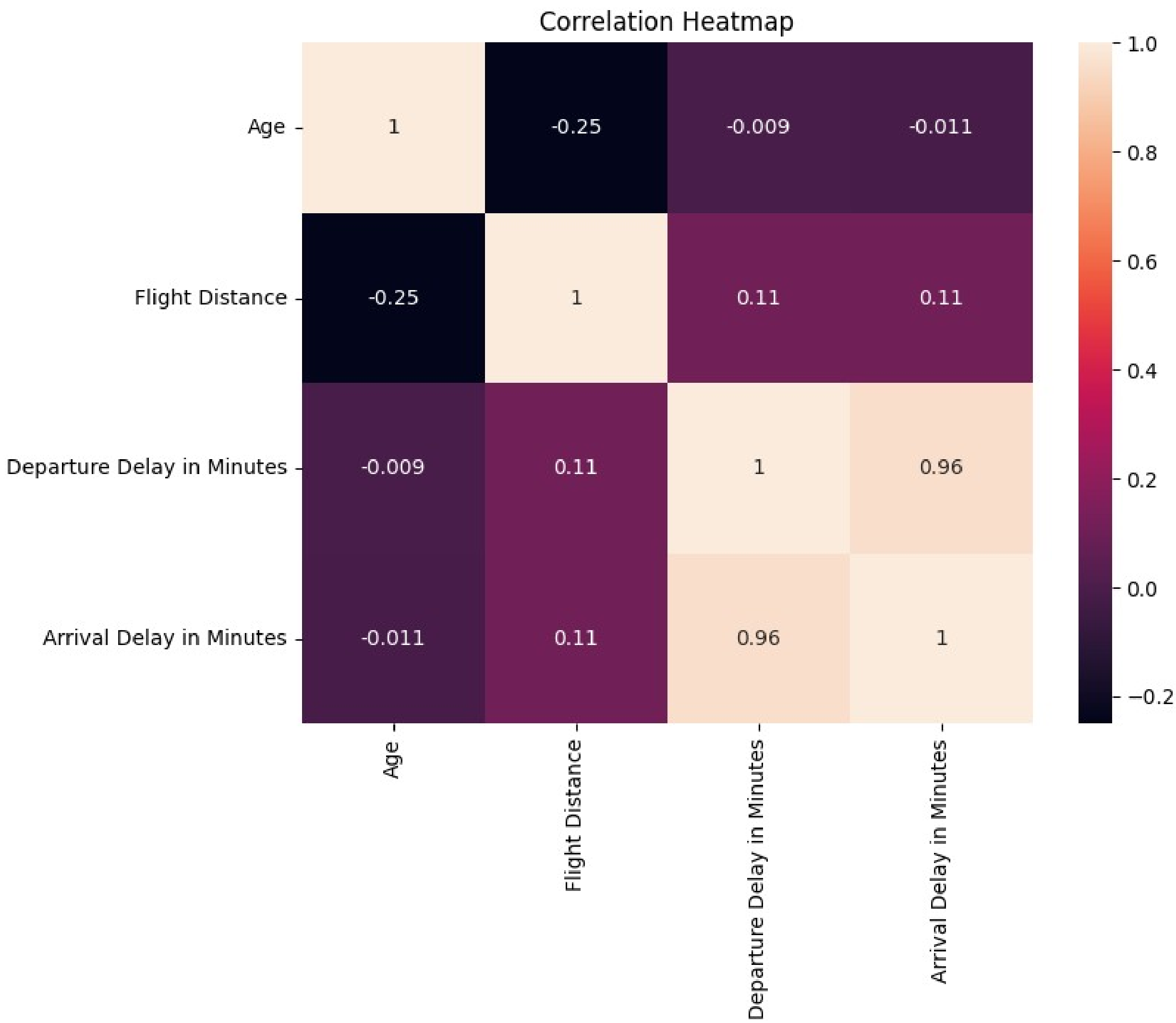
```
#Scatter Plot
plt.figure(figsize=(10,5))
sns.scatterplot(x='Flight Distance',y='Departure Delay in
Minutes',data=df,hue='satisfaction')
plt.title('Flight Distance vs. Departure Delay')
plt.show()
```



- When the flight distance is less than 1500 km, there is either no delay or the delay is relatively minimal compared to longer distances.As the delay increases for longer distances, the dissatisfaction among passengers also tends to rise.

```
#Heatmap
#for Correlation heatmap we need to select numerical data variables
from the dataset ['Age','Flight Distance','Departure Delay in
Minutes','Arrival Delay in Minutes'],the all other features are the
rating features.

numerical_features=['Age','Flight Distance','Departure Delay in
Minutes','Arrival Delay in Minutes']
plt.figure(figsize=(8,6))
correlation_matrix = df[numerical_features].corr()
sns.heatmap(correlation_matrix,annot=True)
plt.title('Correlation Heatmap')
plt.show()
```

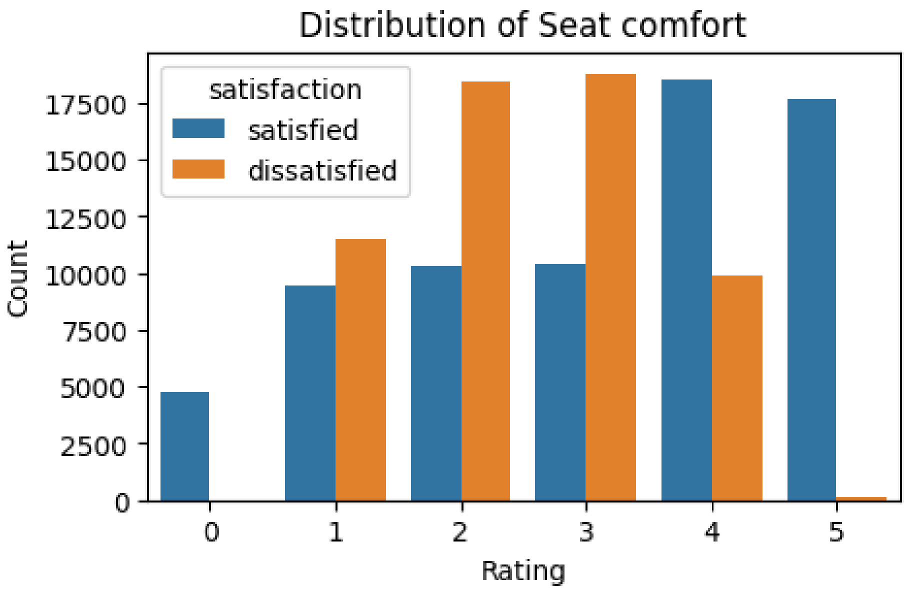


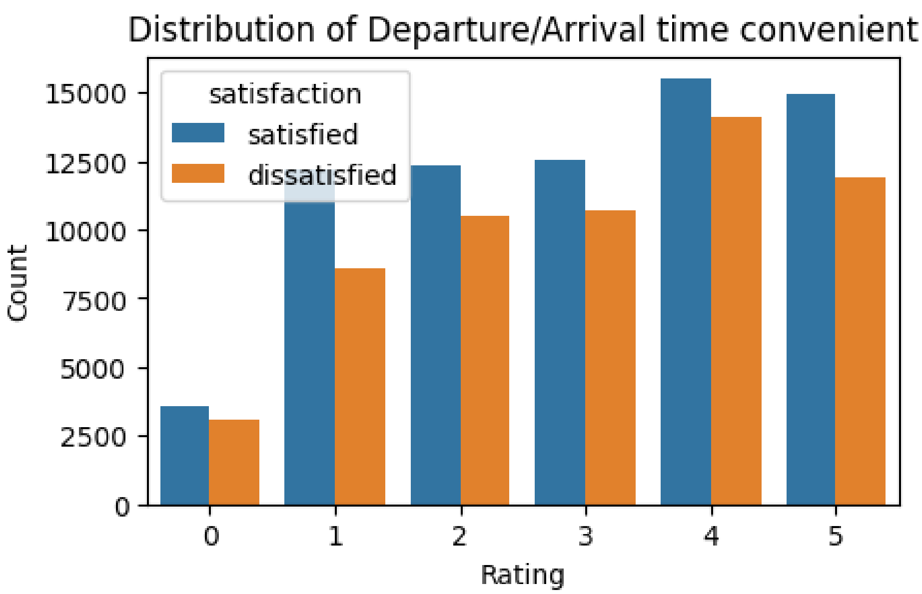
• From the heatmap it is understood that there is a strong positive correlation betweeb Departure Delay and Arrival Delay.This means that if the flight is delay in depature it will affect the arrival also.

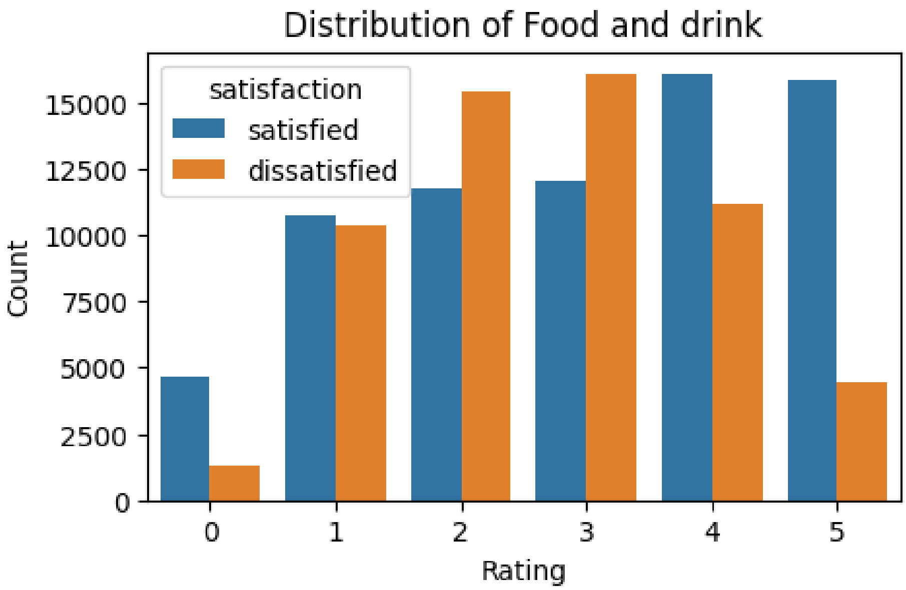
• Departure Delay column and the Arrival Delay column gives similar informations,so we can take one of these columns for further analysis.

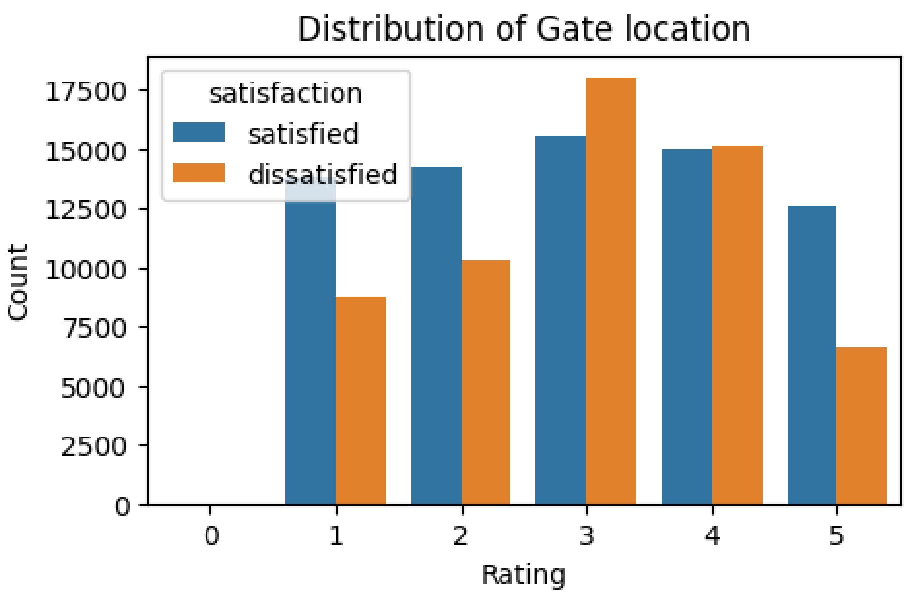
```
# Define the rating columns to visualize
rating_columns = ['Seat comfort', 'Departure/Arrival time convenient','Food and drink',
                  'Gate location','Inflight wifi service', 'Inflight entertainment','Online support',
                  'Ease of Online booking','On-board service', 'Leg room service', 'Baggage handling',
                  'Checkin service', 'Cleanliness','Online boarding']
for c in rating_columns:
    plt.figure(figsize=(5, 3))
```

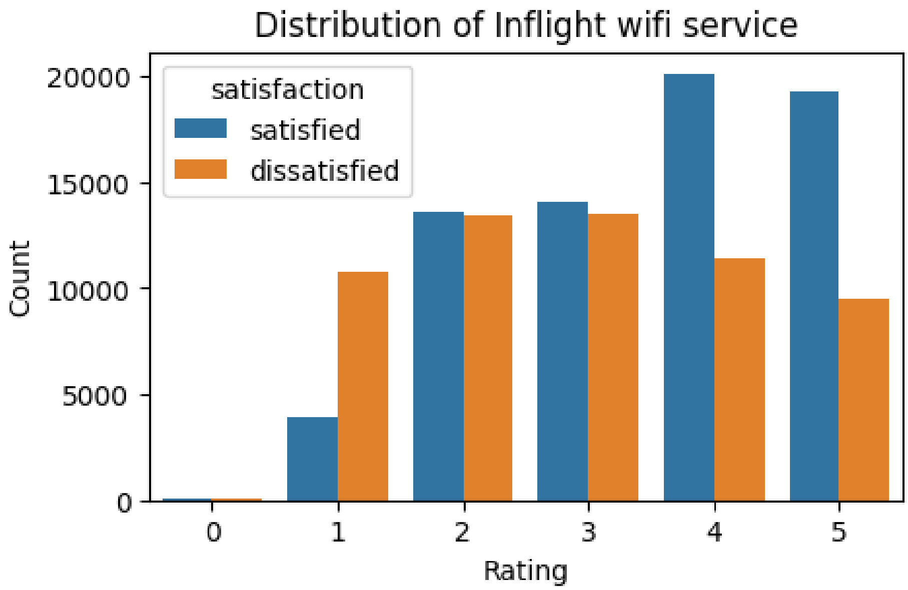
```
    sns.countplot(data=df, x=c, hue='satisfaction')
    plt.title('Distribution of {}'.format(c))
    plt.xlabel('Rating')
    plt.ylabel('Count')
    plt.show()
```

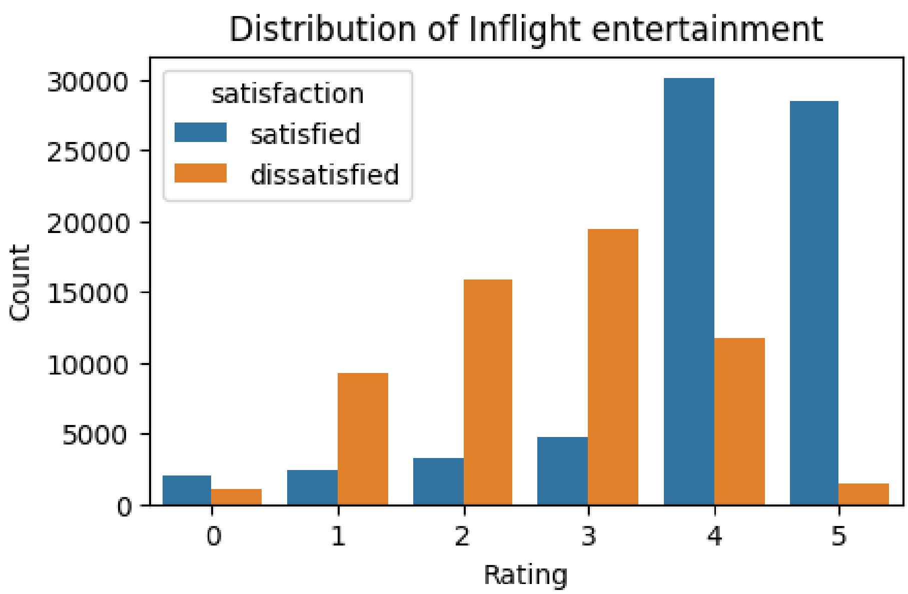


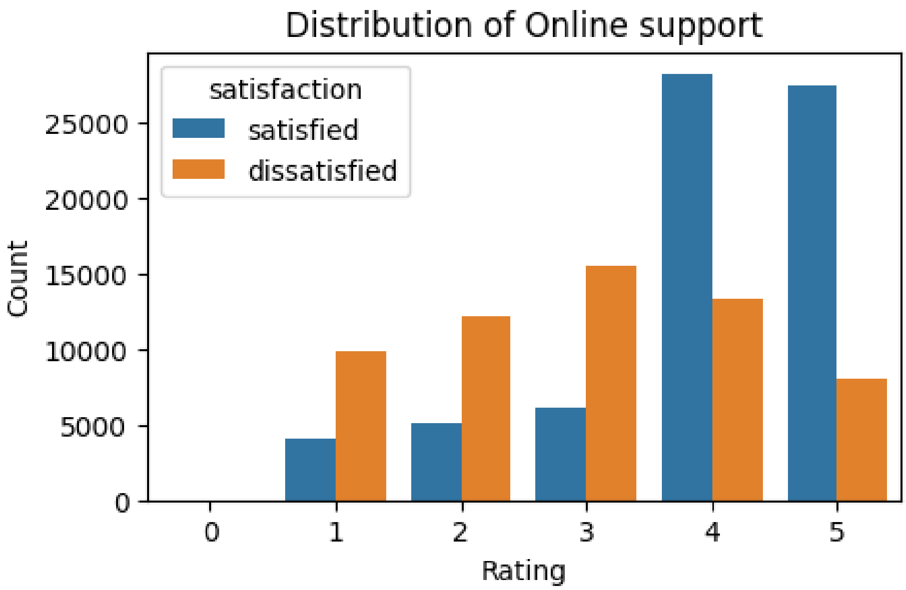


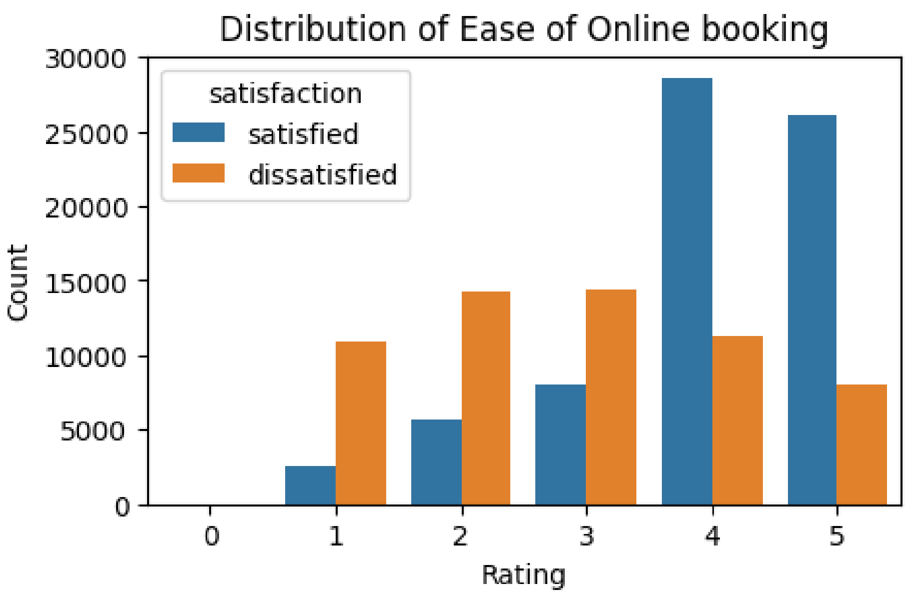

Distribution of Food and drink
satisfaction
satisfied
dissatisfied
Count
15000
12500
10000
7500
5000
2500
0
0
1
2
3
4
5
Rating

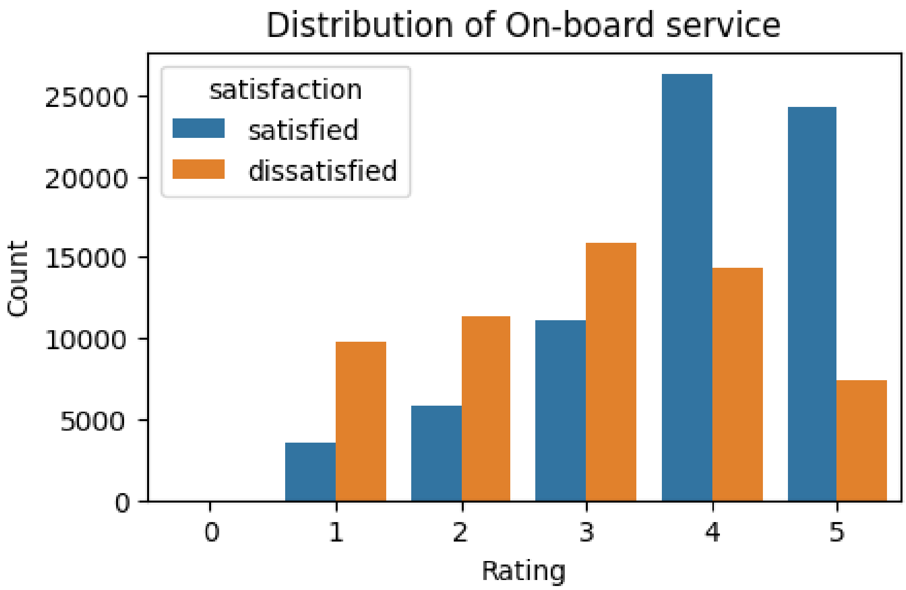

Distribution of Gate location
satisfaction
satisfied
dissatisfied
Count
17500
15000
12500
10000
7500
5000
2500
0
0
1
2
3
4
5
Rating


Distribution of Inflight wifi service
satisfaction
satisfied
dissatisfied
Count
20000
15000
10000
5000
0
0
1
2
3
4
5
Rating

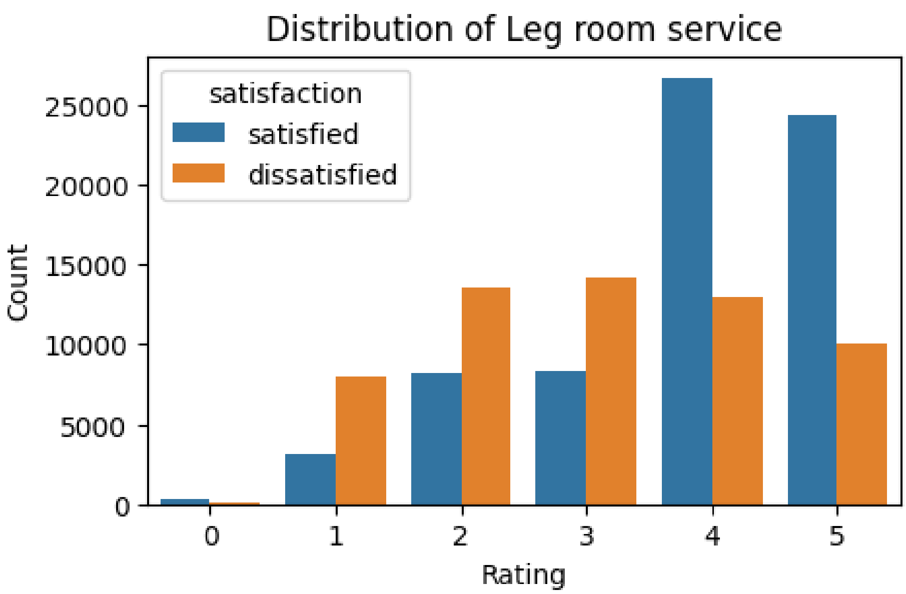

Distribution of Inflight entertainment
satisfaction
satisfied
dissatisfied
Count
30000
25000
20000
15000
10000
5000
0
0
1
2
3
4
5
Rating


Distribution of Online support
satisfaction
satisfied
dissatisfied
Count
Rating
0
5000
10000
15000
20000
25000
0
1
2
3
4
5

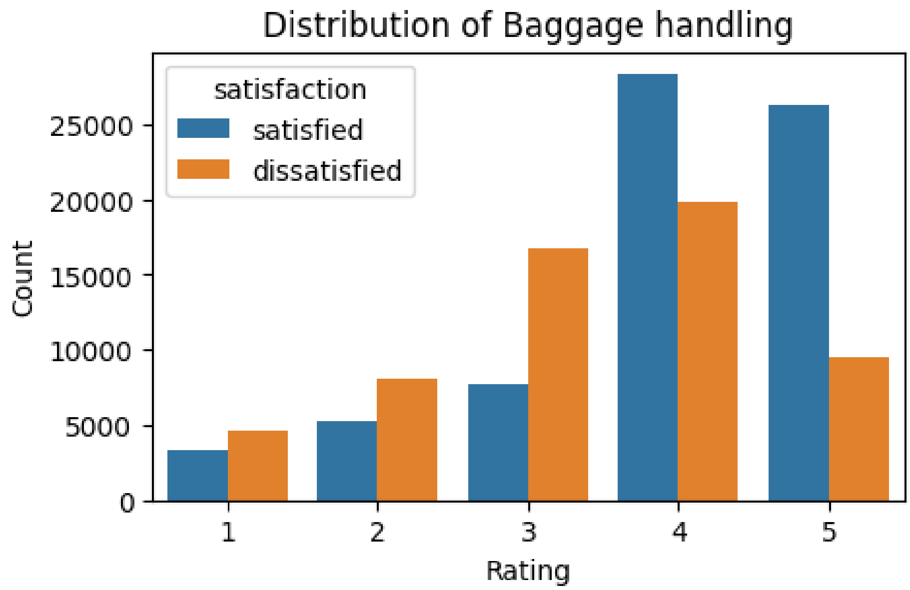

Distribution of Ease of Online booking
satisfaction
satisfied
dissatisfied
Count
Rating
0
5000
10000
15000
20000
25000
30000
0
1
2
3
4
5



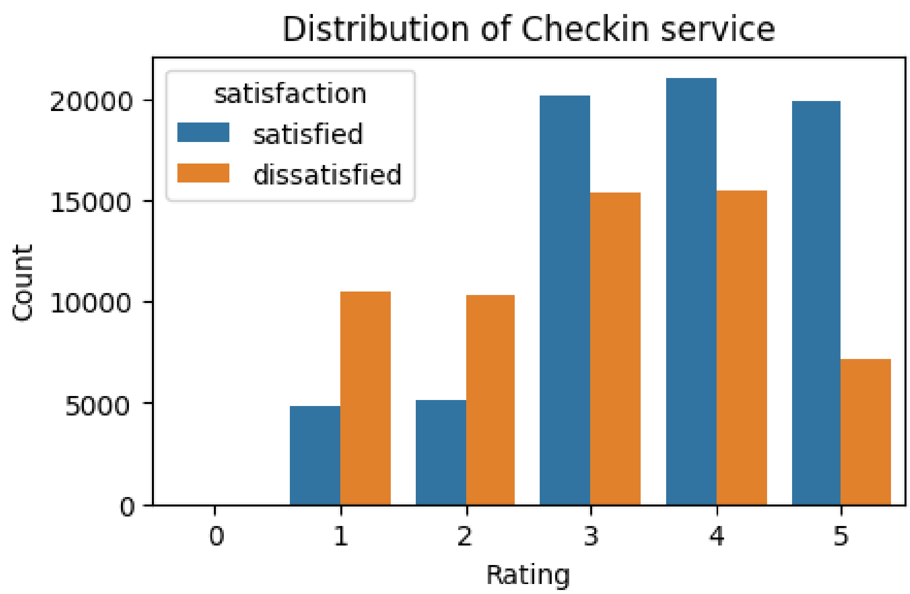
Distribution of On-board service
satisfaction
satisfied
dissatisfied
Count
25000
20000
15000
10000
5000
0
0
1
2
3
4
5
Rating



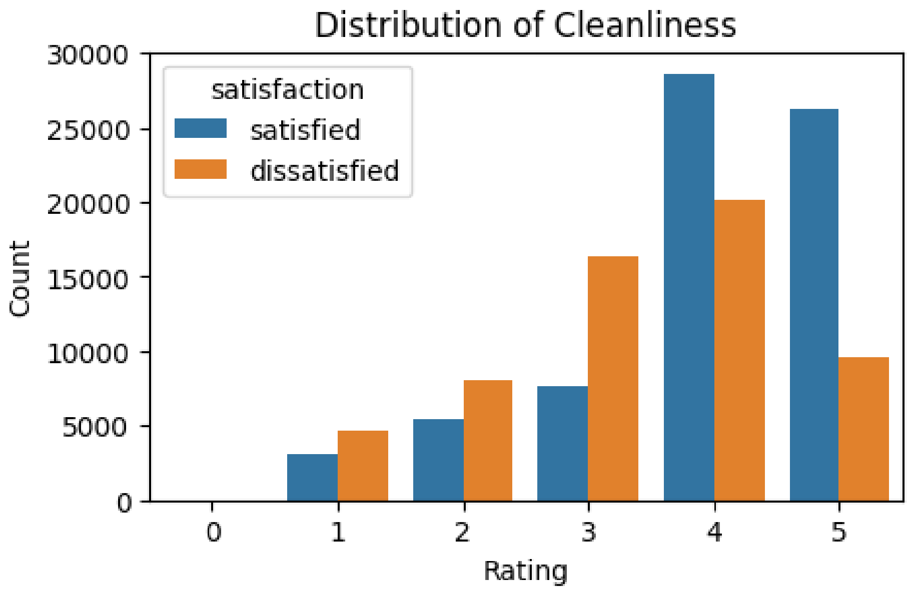
Distribution of Leg room service
satisfaction
satisfied
dissatisfied
Count
25000
20000
15000
10000
5000
0
0
1
2
3
4
5
Rating

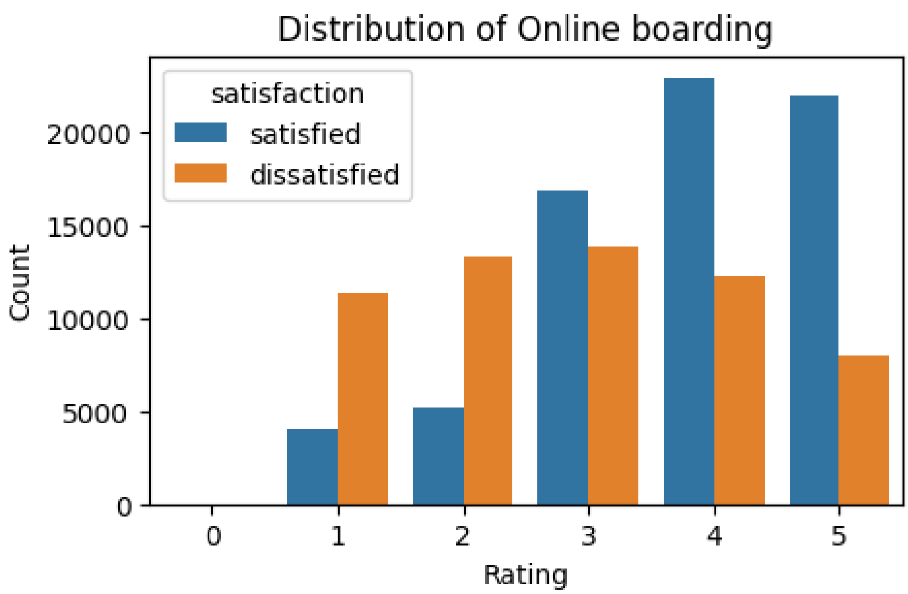


Distribution of Baggage handling
satisfaction
satisfied
dissatisfied
Count
0
5000
10000
15000
20000
25000
1
2
3
4
5
Rating



Distribution of Checkin service
satisfaction
satisfied
dissatisfied
Count
0
5000
10000
15000
20000
0
1
2
3
4
5
Rating





- Customers provide higher ratings for factors such as seat comfort, food and drink,inflight entertainment, and other services, it is associated with a greater likelihood of satisfaction.

- This suggests that customers who express more positive evaluations for these services tend to have a more fulfilling overall experience.

Data Preprocessing

```
#Drop unnecessary columns
# Drop the 'Arrival Delay in Minutes' column from the dataframe
df=df.drop('Arrival Delay in Minutes', axis=1)
```

```
#Use one hot encode to convert the object type features into integer
type
data= pd.get_dummies(df,columns=['Gender','Customer
Type','Class','Type of Travel'],drop_first=True)
data.head()
```

```
  satisfaction  Age  Flight Distance  Seat comfort  \
0    satisfied   65              265             0
1    satisfied   47             2464             0
2    satisfied   15             2138             0
3    satisfied   60              623             0
4    satisfied   70              354             0

   Departure/Arrival time convenient  Food and drink  Gate location  \
0                                  0               0              2
1                                  0               0              3
2                                  0               0              3
3                                  0               0              3
4                                  0               0              3

   Inflight wifi service  Inflight entertainment  Online support  ...
\
0                      2                       4               2  ...

1                      0                       2               2  ...

2                      2                       0               2  ...

3                      3                       4               3  ...

4                      4                       3               4  ...


   Baggage handling  Checkin service  Cleanliness  Online boarding  \
0                 3                5            3                2
1                 4                2            3                2
2                 4                4            4                2
3                 1                4            1                3
4                 2                4            2                5

   Departure Delay in Minutes  Gender_Male  Customer Type_disloyal
Customer  \
0                           0        False
False
1                         310         True
False
2                           0        False
False
3                           0        False
False
```

```
4                           0        False
False

   Class_Eco  Class_Eco Plus  Type of Travel_Personal Travel
0       True           False                            True
1      False           False                            True
2       True           False                            True
3       True           False                            True
4       True           False                            True

[5 rows x 23 columns]
```

```python
#Use the LabelEncoder to the output column
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data['satisfaction'] =le.fit_transform(df['satisfaction'])
data.head()
```

```
   satisfaction  Age  Flight Distance  Seat comfort  \
0             1   65              265             0
1             1   47             2464             0
2             1   15             2138             0
3             1   60              623             0
4             1   70              354             0

   Departure/Arrival time convenient  Food and drink  Gate location  \
0                                  0               0              2
1                                  0               0              3
2                                  0               0              3
3                                  0               0              3
4                                  0               0              3

   Inflight wifi service  Inflight entertainment  Online support  ...
\
0                      2                       4               2  ...

1                      0                       2               2  ...

2                      2                       0               2  ...

3                      3                       4               3  ...

4                      4                       3               4  ...


   Baggage handling  Checkin service  Cleanliness  Online boarding  \
0                 3                5            3                2
1                 4                2            3                2
2                 4                4            4                2
3                 1                4            1                3
4                 2                4            2                5
```

```
   Departure Delay in Minutes  Gender_Male  Customer Type_disloyal
Customer  \
0                           0        False
False
1                         310         True
False
2                           0        False
False
3                           0        False
False
4                           0        False
False

   Class_Eco  Class_Eco Plus  Type of Travel_Personal Travel
0       True           False                            True
1      False           False                            True
2       True           False                            True
3       True           False                            True
4       True           False                            True

[5 rows x 23 columns]
```

```
# Split the dataframe into features (X) and target variable (y)
x = data.drop(['satisfaction'],axis=1)
y= data.satisfaction
x.head()
```

```
   Age  Flight Distance  Seat comfort  Departure/Arrival time
convenient  \
0   65              265             0
0
1   47             2464             0
0
2   15             2138             0
0
3   60              623             0
0
4   70              354             0
0

   Food and drink  Gate location  Inflight wifi service  \
0               0              2                      2
1               0              3                      0
2               0              3                      2
3               0              3                      3
4               0              3                      4

   Inflight entertainment  Online support  Ease of Online booking  ...
\
```

```
0                        4               2                        3  ...

1                        2               2                        3  ...

2                        0               2                        2  ...

3                        4               3                        1  ...

4                        3               4                        2  ...


   Baggage handling  Checkin service  Cleanliness  Online boarding  \
0                 3                5            3                2
1                 4                2            3                2
2                 4                4            4                2
3                 1                4            1                3
4                 2                4            2                5

   Departure Delay in Minutes  Gender_Male  Customer Type_disloyal
Customer  \
0                           0        False
False
1                         310         True
False
2                           0        False
False
3                           0        False
False
4                           0        False
False

   Class_Eco  Class_Eco Plus  Type of Travel_Personal Travel
0       True           False                            True
1      False           False                            True
2       True           False                            True
3       True           False                            True
4       True           False                            True

[5 rows x 22 columns]
```

Standardization

```
from sklearn.preprocessing import StandardScaler
std= StandardScaler()
selected_features = x[["Age", "Flight Distance", "Departure Delay in
Minutes"]]
std.fit(selected_features)
s = std.transform(selected_features)
```

```
x[["Age", "Flight Distance", "Departure Delay in Minutes"]]=s
x.head()
```

```
        Age  Flight Distance  Seat comfort  Departure/Arrival time
convenient  \
0  1.691351        -1.671103             0
0
1  0.500820         0.469852             0
0
2 -1.615680         0.152458             0
0
3  1.360648        -1.322552             0
0
4  2.022054        -1.584452             0
0

   Food and drink  Gate location  Inflight wifi service  \
0               0              2                      2
1               0              3                      0
2               0              3                      2
3               0              3                      3
4               0              3                      4

   Inflight entertainment  Online support  Ease of Online booking  ...
\
0                       4               2                       3  ...

1                       2               2                       3  ...

2                       0               2                       2  ...

3                       4               3                       1  ...

4                       3               4                       2  ...


   Baggage handling  Checkin service  Cleanliness  Online boarding  \
0                 3                5            3                2
1                 4                2            3                2
2                 4                4            4                2
3                 1                4            1                3
4                 2                4            2                5

   Departure Delay in Minutes  Gender_Male  Customer Type_disloyal
Customer  \
0                   -0.386481        False
False
1                    7.756204         True
False
2                   -0.386481        False
```

```
False
3                    -0.386481        False
False
4                    -0.386481        False
False

   Class_Eco  Class_Eco Plus  Type of Travel_Personal Travel
0       True           False                            True
1      False           False                            True
2       True           False                            True
3       True           False                            True
4       True           False                            True

[5 rows x 22 columns]
```

Splitting Data into Training and Testing Dataset

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.20,rando
m_state=0)
x_train.shape

(103904, 22)

x_test.shape

(25976, 22)

x_train

             Age  Flight Distance  Seat comfort  \
24986   0.236257        -1.856087             1
52096  -0.954274        -0.814331             3
114671  0.963804         0.291683             5
76726  -0.028305         1.438588             2
89451   0.236257         0.460116             5
...          ...              ...           ...
45891   0.368538        -0.114310             2
117952  0.831523        -0.925322             3
42613   0.500820         0.111566             1
43567  -0.094446         0.135906             1
68268  -1.020415         3.123897             1

        Departure/Arrival time convenient  Food and drink  Gate
location  \
24986                                  1               1
1
52096                                  2               3
4
```

```
114671                                  5               5
5
76726                                   5               5
5
89451                                   5               5
5
...                                   ...             ...            .
..
45891                                   2               2
1
117952                                  3               3
3
42613                                   1               1
3
43567                                   2               1
3
68268                                   5               5
5

        Inflight wifi service  Inflight entertainment  Online support
\
24986                       5                       5               5

52096                       4                       3               4

114671                      5                       5               4

76726                       5                       4               3

89451                       2                       3               5

...                       ...                     ...             ...

45891                       5                       2               5

117952                      5                       4               5

42613                       2                       1               2

43567                       2                       1               2

68268                       1                       1               1


        Ease of Online booking  ...  Baggage handling  Checkin service
\
24986                        5  ...                 5                5

52096                        4  ...                 4                1

114671                       5  ...                 5                5
```

```
76726                           2  ...                      2                 3
89451                           4  ...                      4                 5
...                           ...  ...                    ...               ...
45891                           5  ...                      5                 4
117952                          5  ...                      5                 5
42613                           2  ...                      3                 2
43567                           2  ...                      4                 2
68268                           1  ...                      3                 3

        Cleanliness  Online boarding  Departure Delay in Minutes  
Gender_Male  \
24986             5                5                   -0.386481   
False   
52096             3                4                    0.243920   
False   
114671            5                3                   -0.386481   
True   
76726             2                3                   -0.386481   
False   
89451             4                3                    0.848055   
True   
...             ...              ...                         ...   
...   
45891             5                5                    0.664188   
True   
117952            5                5                   -0.386481   
False   
42613             4                2                    0.270187   
False   
43567             3                2                   -0.386481   
True   
68268             4                1                   -0.255147   
True   

        Customer Type_disloyal Customer  Class_Eco  Class_Eco Plus  \
24986                             False       True           False  
52096                              True       True           False  
114671                            False      False           False  
76726                             False      False           False  
89451                             False      False           False  
...                                 ...        ...             ...  
```

```
45891                                  True      False           False
117952                                False      False           False
42613                                  True       True           False
43567                                  True       True           False
68268                                 False      False           False

        Type of Travel_Personal Travel
24986                             True
52096                            False
114671                           False
76726                            False
89451                            False
...                                ...
45891                            False
117952                           False
42613                            False
43567                            False
68268                            False

[103904 rows x 22 columns]
```

x_test

```
             Age  Flight Distance  Seat comfort  \
125669  0.699242        -0.745205             5
90648   0.699242        -1.084019             4
45322  -0.160587        -0.623505             2
64084   0.037835        -1.667208             0
71595  -0.094446         1.437615             2
...          ...              ...           ...
52679   0.633101        -0.374262             3
95042   0.699242        -1.307948             4
16588  -0.028305         0.010311             2
98261   1.029945         1.759878             5
19790   0.897663         0.962496             2

        Departure/Arrival time convenient  Food and drink  Gate
location  \
125669                                  3               3
3
90648                                   4               4
4
45322                                   1               1
2
64084                                   0               0
2
71595                                   1               1
1
...                                   ...             ...             .
..
```

```
52679                                  3              3
2
95042                                  4              2
4
16588                                  5              2
1
98261                                  5              5
5
19790                                  5              3
4

        Inflight wifi service  Inflight entertainment  Online support
\
125669                      5                       5               5

90648                       2                       5               5

45322                       4                       1               4

64084                       4                       5               4

71595                       1                       2               2

...                       ...                     ...             ...

52679                       1                       3               1

95042                       4                       4               5

16588                       5                       2               5

98261                       4                       5               5

19790                       4                       3               3


        Ease of Online booking  ...  Baggage handling  Checkin service
\
125669                       5  ...                 1                2

90648                        4  ...                 4                2

45322                        4  ...                 5                5

64084                        4  ...                 4                3

71595                        2  ...                 2                1

...                        ...  ...               ...              ...

52679                        1  ...                 3                4
```

```
95042                        4  ...                    4                 3
16588                        5  ...                    2                 2
98261                        4  ...                    4                 5
19790                        4  ...                    5                 4

        Cleanliness  Online boarding  Departure Delay in Minutes
Gender_Male  \
125669            3                5                   -0.386481
True
90648             4                5                   -0.386481
True
45322             5                4                   -0.386481
False
64084             4                4                    0.138853
True
71595             2                2                   -0.386481
False
...             ...              ...                         ...
...
52679             2                1                   -0.333948
True
95042             4                5                   -0.386481
False
16588             5                5                    0.033787
True
98261             4                4                   -0.281414
False
19790             4                4                   -0.018747
True

        Customer Type_disloyal Customer  Class_Eco  Class_Eco Plus  \
125669                            False      False            True
90648                             False      False           False
45322                              True      False           False
64084                             False      False           False
71595                             False      False           False
...                                 ...        ...             ...
52679                              True      False            True
95042                             False      False           False
16588                             False       True           False
98261                             False      False           False
19790                             False       True           False

        Type of Travel_Personal Travel
125669                           False
```

```
90648                               False
45322                               False
64084                               False
71595                               False
...                                   ...
52679                               False
95042                               False
16588                                True
98261                               False
19790                                True

[25976 rows x 22 columns]
```

MODEL TRAINING

Classification Algorithms

1. Naive Bayes
2. Support vector Machine
3. Decision Tree
4. Random Forest
5. Ada Boost
6. XG Boost
7. Logistic Regression

```
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import BernoulliNB
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from xgboost import XGBClassifier
from sklearn.linear_model import LogisticRegression
```

Perfomance evaluation

1. Confusion Matrix: It is a matrix of size 2×2 for binary classification with actual values on one axis and predicted on another.

2. Accuracy Score: Accuracy is the measure of correct predictions made by our model.It is equal to the number of correct predictions made upon total number of predictions made by the model.

3. Precision Score: It is defined as the ratio of true positives to the sum of true andfalse positives. It is also known as Positive Predictive Value (PPV).

4. Recall Score: It is defined as the ratio of true positives to the sum of true positives and false negatives. It is also called True Positive Rate (TPR) or sensitivity.

5. F1 score: It is the weighted harmonic mean of precision and recall. The closer the value of the F1 score is to 1.0 , the better the expected performance of the model is.

1A). Naive Bayes:GaussianNB

```
nb = GaussianNB()
nb.fit(x_train,y_train)
y_pred2=nb.predict(x_test)
y_pred2
```

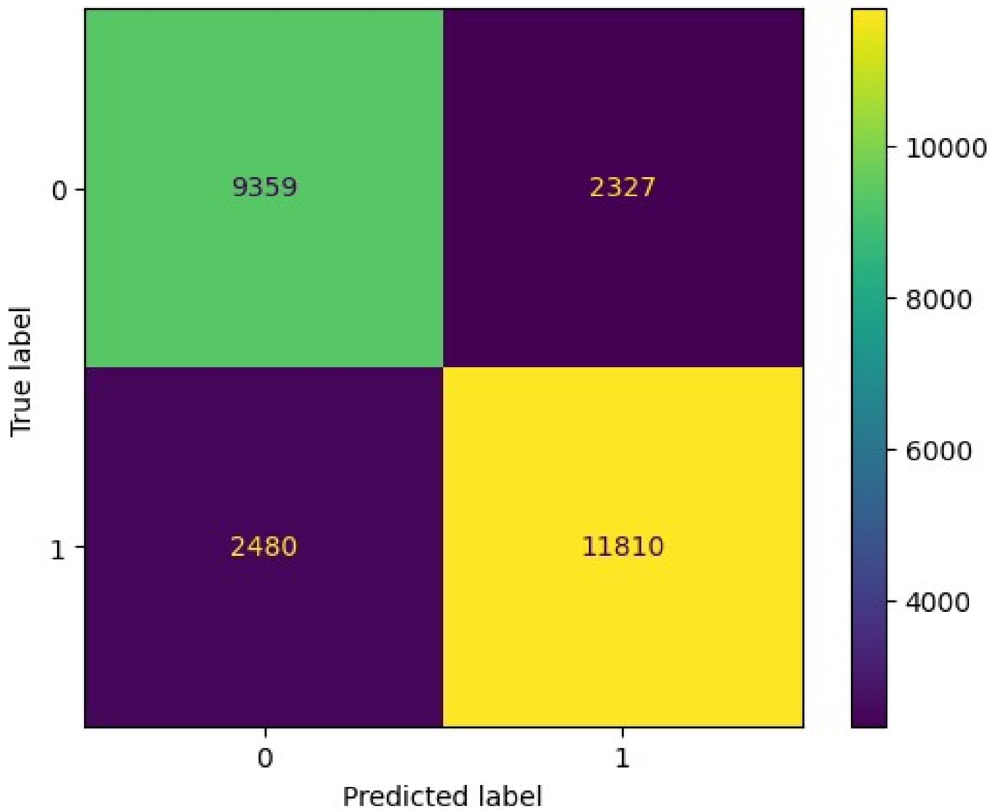
```
array([1, 1, 1, ..., 1, 1, 1])
```

```
#confusion matrix
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
result = confusion_matrix(y_test,y_pred2)
print(result)
labels=[0,1]
cmd = ConfusionMatrixDisplay(result,display_labels=labels)
```

```
[[ 9359  2327]
 [ 2480 11810]]
```

```
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x18eaeea5d50>
```




```
# Accuracy score and classification report
from sklearn.metrics import accuracy_score,classification_report
print('Accuracy: ',accuracy_score(y_test,y_pred2)*100,'\n')
print(classification_report(y_test,y_pred2))
```

```
Accuracy:  81.49445642131198

              precision    recall  f1-score   support

           0       0.79      0.80      0.80     11686
           1       0.84      0.83      0.83     14290

    accuracy                           0.81     25976
   macro avg       0.81      0.81      0.81     25976
weighted avg       0.82      0.81      0.82     25976
```

```
training_score= nb.score(x_train,y_train)
training_score
```

```
0.8213158299969202
```

1B). Naive Bayes:BernoulliNB

```
nb_model2=BernoulliNB()
nb_model2.fit(x_train,y_train)
y_pred3=nb_model2.predict(x_test)
y_pred3
```

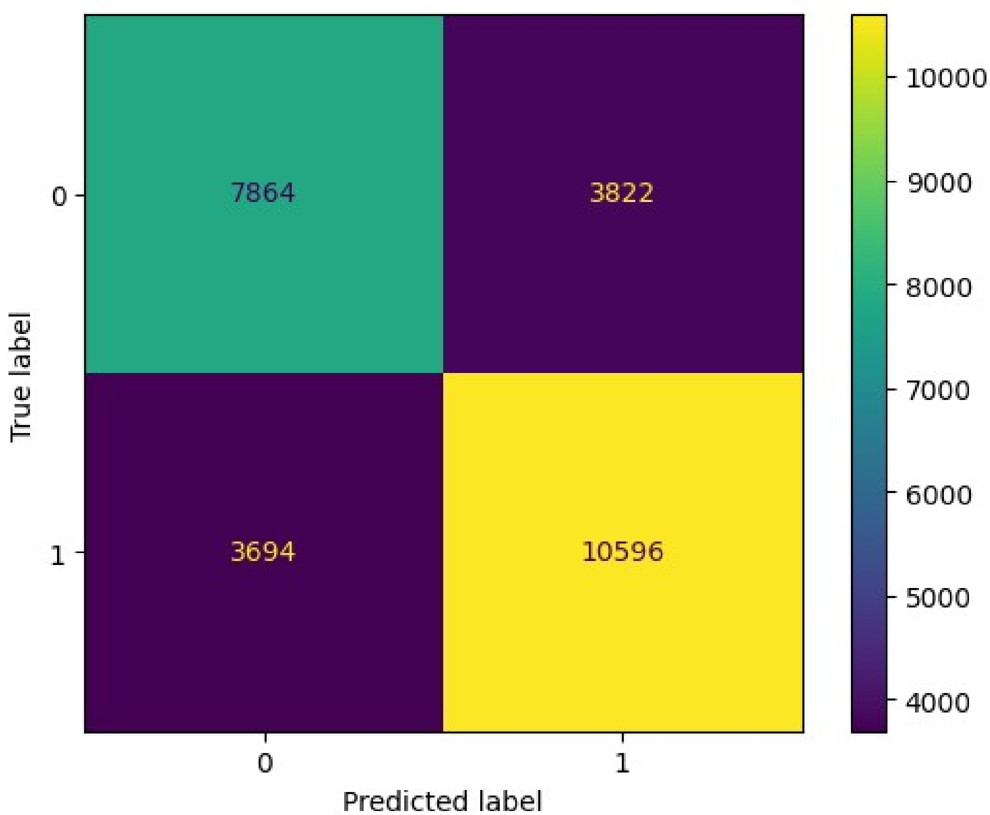
```
array([1, 1, 0, ..., 0, 1, 0])
```

```
#confusion matrix
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
result = confusion_matrix(y_test,y_pred3)
print(result)
labels=[0,1]
cmd = ConfusionMatrixDisplay(result,display_labels=labels)
```

```
[[ 7864  3822]
 [ 3694 10596]]
```

```
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x18eafb34b50>
```




```
# Accuracy score and classification report
from sklearn.metrics import accuracy_score,classification_report
```

```
print('Accuracy: ',accuracy_score(y_test,y_pred3)*100,'\n')
print(classification_report(y_test,y_pred3))
```

```
Accuracy:  71.0655990144749

              precision    recall  f1-score   support

           0       0.68      0.67      0.68     11686
           1       0.73      0.74      0.74     14290

    accuracy                           0.71     25976
   macro avg       0.71      0.71      0.71     25976
weighted avg       0.71      0.71      0.71     25976
```

```
training_score= nb_model2.score(x_train,y_train)
training_score
```

```
0.7161321989528796
```

```
testing_score =nb_model2.score(x_test,y_test)
testing_score
```

```
0.710655990144749
```

2) Support Vector Machine Algorithm(SVM)

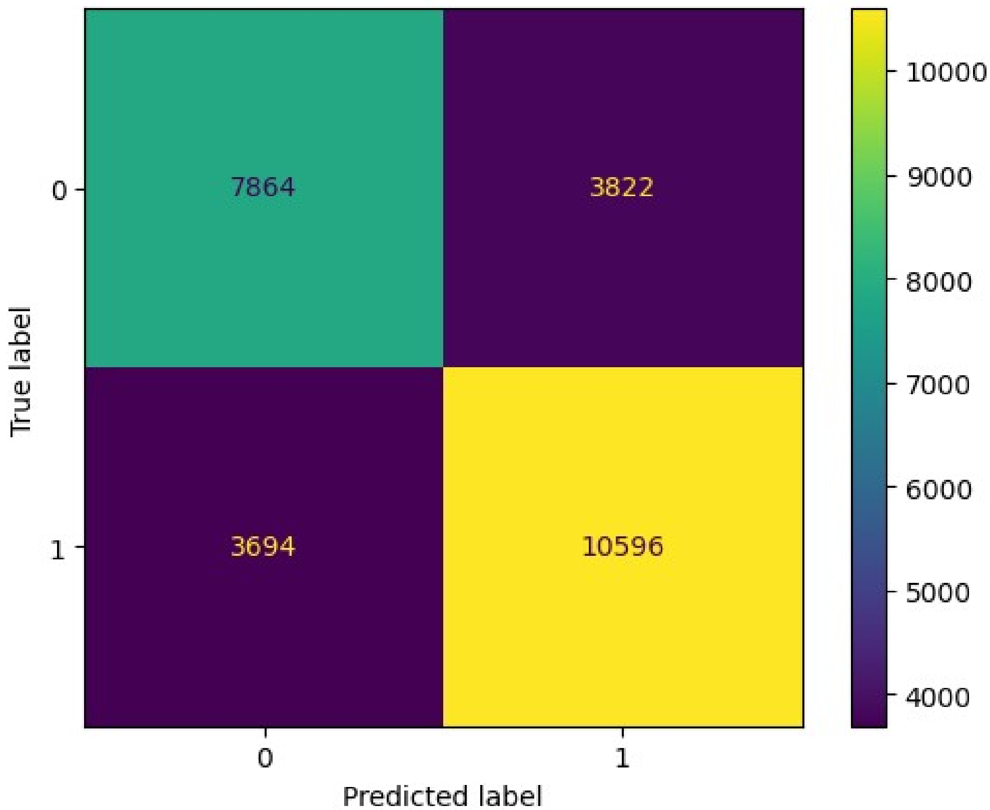
Here, Machine Learning models learn from the past input data and predict the output. Support vector machines are basically supervised learning models used for classification and regression analysis.

```
svc=SVC()
svc.fit(x_train,y_train)
y_pred4=svc.predict(x_test)
y_pred4
```

```
array([1, 1, 0, ..., 0, 1, 0])
```

```
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x18eafb34b50>
```




```
# Accuracy score and classification report
from sklearn.metrics import accuracy_score,classification_report
print('Accuracy: ',accuracy_score(y_test,y_pred4)*100,'\n')
print(classification_report(y_test,y_pred4))

Accuracy:  92.90498922081922

              precision    recall  f1-score   support

           0       0.92      0.93      0.92     11686
           1       0.94      0.93      0.94     14290

    accuracy                           0.93     25976
   macro avg       0.93      0.93      0.93     25976
weighted avg       0.93      0.93      0.93     25976


training_score= svc.score(x_train,y_train)
training_score

0.9317254388666462

testing_score = svc.score(x_test,y_test)
testing_score

0.9290498922081922
```

3) Decision tree

A decision tree is one of the most powerful tools of supervised learning algorithms used for both classification and regression tasks. It builds a flowchart-like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

```
dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
y_pred5 = dt.predict(x_test)
y_pred5
```

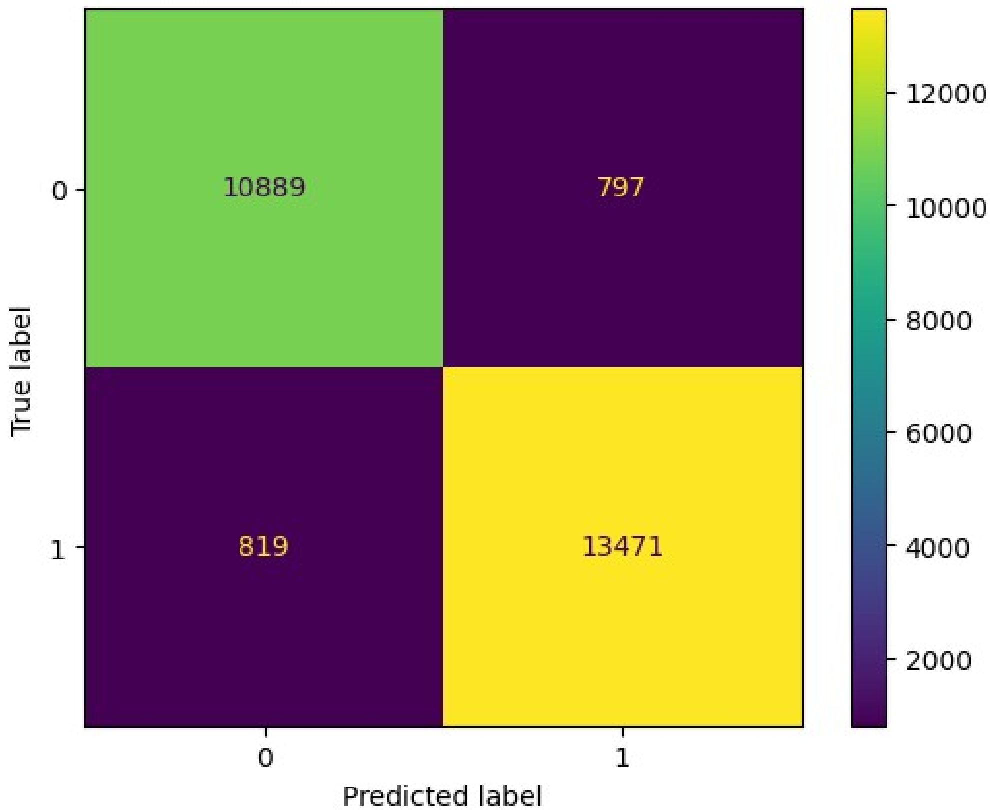
```
array([1, 1, 0, ..., 0, 1, 0])
```

```
#confusion matrix
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
result = confusion_matrix(y_test,y_pred5)
print(result)
labels=[0,1]
cmd = ConfusionMatrixDisplay(result,display_labels=labels)
```

```
[[10889   797]
 [  819 13471]]
```

```
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x18ebba6c550>
```




```
# Accuracy score and classification report
from sklearn.metrics import accuracy_score,classification_report
print('Accuracy: ',accuracy_score(y_test,y_pred5)*100,'\n')
print(classification_report(y_test,y_pred5))
```

```
Accuracy:  93.77887280566677

              precision    recall  f1-score   support

           0       0.93      0.93      0.93     11686
           1       0.94      0.94      0.94     14290

    accuracy                           0.94     25976
   macro avg       0.94      0.94      0.94     25976
weighted avg       0.94      0.94      0.94     25976
```

```
training_score= dt.score(x_train,y_train)
training_score
```

```
1.0
```

```
testing_score = dt.score(x_test,y_test)
testing_score
```

```
0.9377887280566677
```

4) Random Forest

Random forests or random decision forests is an ensemble learning method for classification,regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.

```
rfc=RandomForestClassifier(n_estimators=20,criterion='gini',max_depth=
5,max_features=5)
rfc.fit(x_train,y_train)
y_pred6 = rfc.predict(x_test)
y_pred6
```

```
array([1, 1, 0, ..., 0, 1, 0])
```

```
#confusion matrix
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
result = confusion_matrix(y_test,y_pred6)
print(result)
labels=[0,1]
cmd = ConfusionMatrixDisplay(result,display_labels=labels)
```

```
[[10104  1582]
 [ 1298 12992]]
```

```
# Accuracy score and classification report
from sklearn.metrics import accuracy_score,classification_report
print('Accuracy: ',accuracy_score(y_test,y_pred6)*100,'\n')
print(classification_report(y_test,y_pred6))
```

```
Accuracy:  88.91284262396057

              precision    recall  f1-score   support

           0       0.89      0.86      0.88     11686
           1       0.89      0.91      0.90     14290

    accuracy                           0.89     25976
   macro avg       0.89      0.89      0.89     25976
weighted avg       0.89      0.89      0.89     25976
```

```
training_score= rfc.score(x_train,y_train)
training_score
```

```
0.8924584231598398
```

```
testing_score = rfc.score(x_test,y_test)
testing_score
```

```
0.8891284262396058
```

5) Ada Boost

Ada Boost is an ensemble learning method. We use boosting for combining weak learners with high bias. Boosting aims to produce a model with a lower bias than that of the individual models.

```
abc_model = AdaBoostClassifier(n_estimators=10,learning_rate=1.0)
abc_model.fit(x_train,y_train)
y_pred7=abc_model.predict(x_test)
y_pred7
```

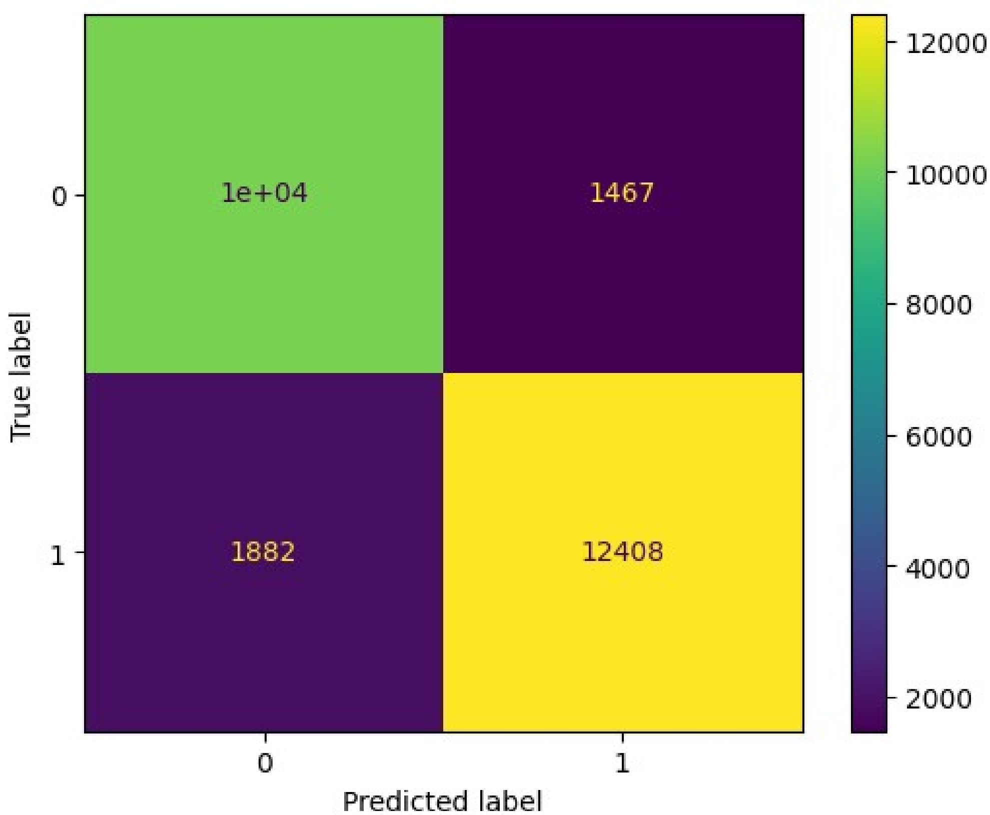
```
array([1, 1, 0, ..., 0, 1, 0])
```

```
#confusion matrix
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
result=confusion_matrix(y_test,y_pred7)
print(result)
labels=[0,1]
cmd=ConfusionMatrixDisplay(result,display_labels=labels)
```

```
[[10219  1467]
 [ 1882 12408]]
```

```
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x18eaf1f11d0>
```



```
#accuracy score and classification report
from sklearn.metrics import accuracy_score,classification_report
print('Accuracy',accuracy_score(y_test,y_pred7)*100,'\n')
print(classification_report(y_test,y_pred7))
```

```
Accuracy 87.10732984293193

              precision    recall  f1-score   support

           0       0.84      0.87      0.86     11686
           1       0.89      0.87      0.88     14290

    accuracy                           0.87     25976
   macro avg       0.87      0.87      0.87     25976
weighted avg       0.87      0.87      0.87     25976
```

```
training_score = abc_model.score(x_train,y_train)
training_score
```

```
0.8728730366492147
```

```
testing_score = abc_model.score(x_test,y_test)
testing_score
```

```
0.8710732984293194
```

6) XG Boost

XG Boost is an ensemble learning method. We use boosting for combining weak learners with high bias. Boosting aims to produce a model with a lower bias than that of the individual models.

```
xgb_model = XGBClassifier()
xgb_model.fit(x_train,y_train)
y_pred8 = xgb_model.predict(x_test)
y_pred8
```

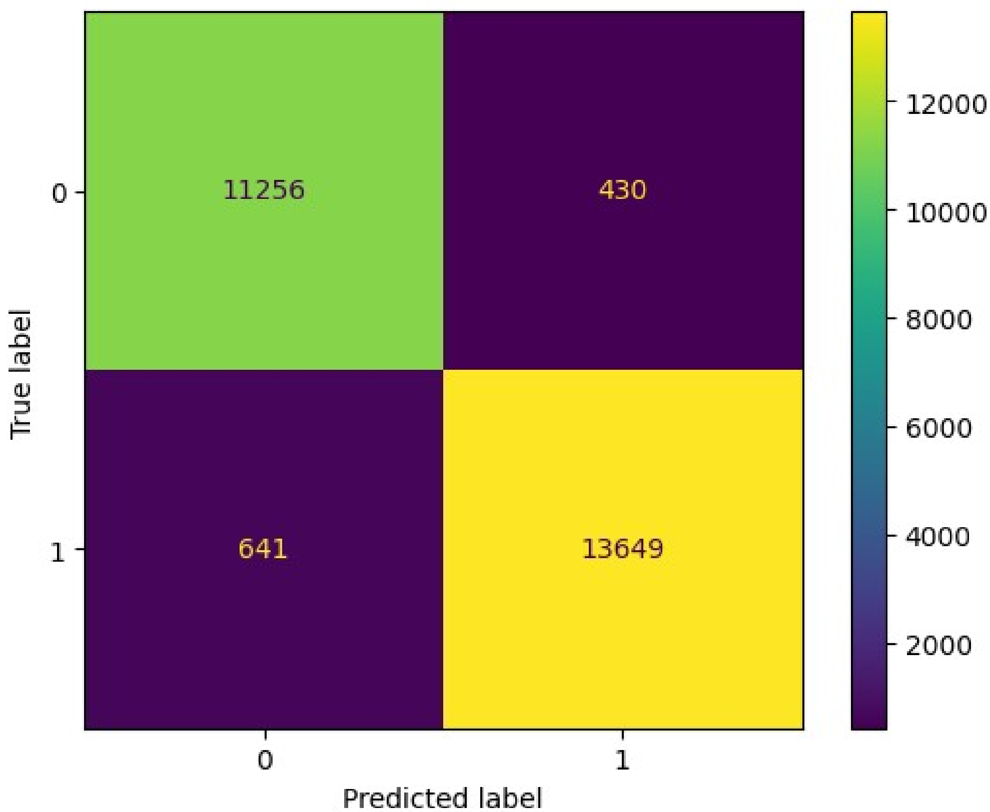
```
array([1, 1, 0, ..., 0, 1, 0])
```

```
#confusion matrix
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
result=confusion_matrix(y_test,y_pred8)
print(result)
labels=[0,1]
cmd=ConfusionMatrixDisplay(result,display_labels=labels)
```

```
[[11256   430]
 [  641 13649]]
```

```
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x18ebbb0c650>
```




```
training_score = xgb_model.score(x_train,y_train)
training_score

0.9704919926085618

testing_score = xgb_model.score(x_test,y_test)
testing_score

0.9587696335078534
```

7) Logistic Regression

It is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

```
lr_model = LogisticRegression()
lr_model.fit(x_train,y_train)
y_pred9 = lr_model.predict(x_test)
y_pred9
```

```
C:\Users\HP\AppData\Local\Programs\Python\Python311\Lib\site-packages\
sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs
failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

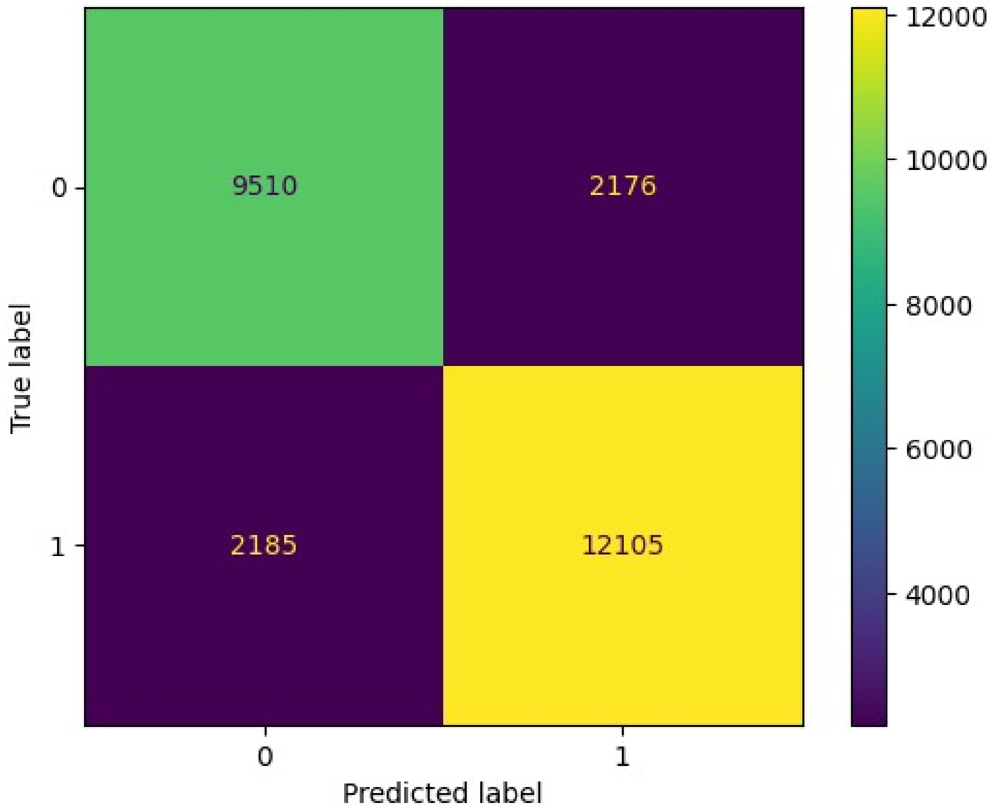
```
array([1, 1, 1, ..., 0, 1, 0])
```

```python
#confusion matrix
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
result=confusion_matrix(y_test,y_pred9)
print(result)
labels=[0,1]
cmd=ConfusionMatrixDisplay(result,display_labels=labels)
```

```
[[ 9510  2176]
 [ 2185 12105]]
```

```python
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x18ebc303590>
```




```
#accuracy score and classification report
from sklearn.metrics import accuracy_score,classification_report
print('Accuracy',accuracy_score(y_test,y_pred9)*100,'\n')
print(classification_report(y_test,y_pred9))
```

```
Accuracy 83.21142593162921

              precision    recall  f1-score   support

           0       0.81      0.81      0.81     11686
           1       0.85      0.85      0.85     14290

    accuracy                           0.83     25976
   macro avg       0.83      0.83      0.83     25976
weighted avg       0.83      0.83      0.83     25976
```

```
training_score = lr_model.score(x_train,y_train)
training_score
```

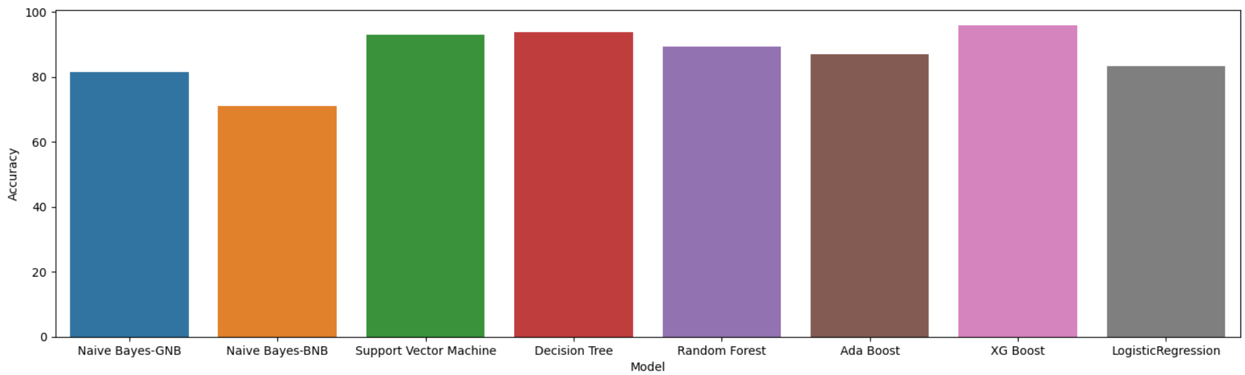
```
0.8367146596858639
```

```
testing_score = lr_model.score(x_test,y_test)
testing_score
```

```
0.832114259316292

#accuracy score comparison
plt.figure(figsize=(18,5))
comparison = pd.DataFrame({'Model':['Naive Bayes-GNB','Naive Bayes-
BNB','Support Vector Machine',
                                    'Decision Tree','Random
Forest','Ada Boost','XG Boost','LogisticRegression'],'Accuracy':
[81.49,71.06,92.90,93.82,89.33,87.10,95.87,83.22]})
sns.barplot(x='Model',y='Accuracy',data=comparison)

<Axes: xlabel='Model', ylabel='Accuracy'>
```



OBSERVATION

- Here, we can see that the XG Boost model has the highest accuracy score (95.87).

```
# Now we need to check the performance of the Balanced dataset
```

OVERSAMPLING

Oversampling is a technique used in machine learning to address class imbalance by increasing the number of instances in the minority class (the less frequent class). This helps to balance the class distribution, which can lead to better model performance.

- SMOTE-for data balancing-synthetic minority over sampling Technique.

```
from imblearn.over_sampling import SMOTE
sm=SMOTE()
x_res,y_res = sm.fit_resample(x,y)
y.value_counts()

satisfaction
1    71087
```

```
0    58793
Name: count, dtype: int64

y_res.value_counts()

satisfaction
1    71087
0    71087
Name: count, dtype: int64

selected_features = x_res[["Age", "Flight Distance", "Departure Delay
in Minutes"]]
std.fit(selected_features)
s = std.transform(selected_features)
x_res[["Age", "Flight Distance", "Departure Delay in Minutes"]]=s
x_res.head()

        Age  Flight Distance  Seat comfort  Departure/Arrival time
convenient  \
0  1.703354        -1.699819             0
0
1  0.512445         0.474676             0
0
2 -1.604728         0.152309             0
0
3  1.372546        -1.345809             0
0
4  2.034163        -1.611811             0
0

   Food and drink  Gate location  Inflight wifi service  \
0               0              2                      2
1               0              3                      0
2               0              3                      2
3               0              3                      3
4               0              3                      4

   Inflight entertainment  Online support  Ease of Online booking  ...
\
0                       4               2                       3  ...

1                       2               2                       3  ...

2                       0               2                       2  ...

3                       4               3                       1  ...

4                       3               4                       2  ...


   Baggage handling  Checkin service  Cleanliness  Online boarding  \
```

```
0                 3                 5             3                2
1                 4                 2             3                2
2                 4                 4             4                2
3                 1                 4             1                3
4                 2                 4             2                5

   Departure Delay in Minutes  Gender_Male  Customer Type_disloyal
Customer  \
0                   -0.386563        False
False
1                    7.691034         True
False
2                   -0.386563        False
False
3                   -0.386563        False
False
4                   -0.386563        False
False

   Class_Eco  Class_Eco Plus  Type of Travel_Personal Travel
0       True           False                            True
1      False           False                            True
2       True           False                            True
3       True           False                            True
4       True           False                            True

[5 rows x 22 columns]
```

```
#splitting
x_train_res,x_test_res,y_train_res,y_test_res =
train_test_split(x_res,y_res,test_size=0.20,random_state=0)
xgb_model = XGBClassifier()
xgb_model.fit(x_train_res,y_train_res)
y_pred = xgb_model.predict(x_test_res)
y_pred
```

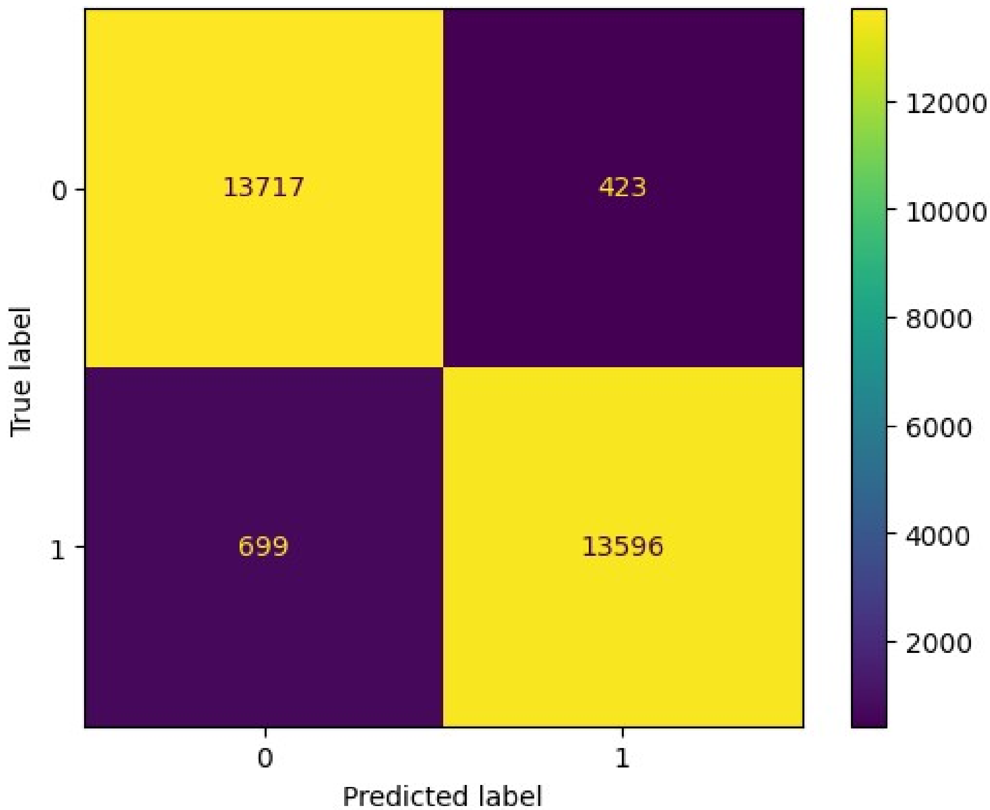
```
array([1, 0, 0, ..., 0, 1, 0])
```

```
#confusion matrix
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
result_os = confusion_matrix(y_test_res,y_pred)
print(result_os)
labels=[0,1]
cmd_os = ConfusionMatrixDisplay(result_os,display_labels=labels)
```

```
[[13717   423]
 [  699 13596]]
```

```
cmd_os.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x18ed449f350>
```




```
#accuracy score and classification report
from sklearn.metrics import accuracy_score,classification_report
print('Accuracy',accuracy_score(y_test_res,y_pred)*100,'\n')
print(classification_report(y_test_res,y_pred))
```

```
Accuracy 96.05415860735009

              precision    recall  f1-score   support

           0       0.95      0.97      0.96     14140
           1       0.97      0.95      0.96     14295

    accuracy                           0.96     28435
   macro avg       0.96      0.96      0.96     28435
weighted avg       0.96      0.96      0.96     28435
```

```
training_score = xgb_model.score(x_train_res,y_train_res)
training_score
```

```
0.9724456870554515
```

```
testing_score = xgb_model.score(x_test_res,y_test_res)
testing_score
```

```
0.9605415860735009
```

OBSERVATION

- Here there is difference between the accuracy of balanced data and imbalanced data.
- so we can choose the balanced data with XGBoost has best model

CONCLUSION

- Here we first fit the model using imbalanced data and we we the accuracy and f1 score corresponding that model.Next we get a better model when we balanced the data . So here balanced data gives better accuracy and f1 score.
- In conclusion, this notebook provides valuable insights into customer satisfaction in the airline industry. Factors such as inflight entertainment, seat comfort, ease of online booking, and online support have a significant impact on satisfaction levels.
- The factors on which airlines needs to focus more is on the 'Arrival Delay in Minutes' and 'Departure Delay in Minutes'
- Based on the analysis, we recommend focusing on improving these areas to enhance customer experiences and increase satisfaction.
- The factors on which airlines needs to focus more is on the 'Arrival Delay in Minutes' and 'Departure Delay in Minutes'