

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [4]: df=pd.read_csv(r"Airline Passenger Satisfaction.csv")
df
```

Out[4]:

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	C
0	0	19556	Female	Loyal Customer	52	Business travel	
1	1	90035	Female	Loyal Customer	36	Business travel	Busi
2	2	12360	Male	disloyal Customer	20	Business travel	
3	3	77959	Male	Loyal Customer	44	Business travel	Busi
4	4	36875	Female	Loyal Customer	49	Business travel	
...	...	...	...	...	...	...	
25971	25971	78463	Male	disloyal Customer	34	Business travel	Busi
25972	25972	71167	Male	Loyal Customer	23	Business travel	Busi
25973	25973	37675	Female	Loyal Customer	17	Personal Travel	
25974	25974	90086	Male	Loyal Customer	14	Business travel	Busi
25975	25975	34799	Female	Loyal Customer	42	Personal Travel	

25976 rows × 25 columns



```
In [5]: df.info()
```

```


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25976 entries, 0 to 25975
Data columns (total 25 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Unnamed: 0                               25976 non-null  int64
1   id                                         25976 non-null  int64
2   Gender                                    25976 non-null  object
3   Customer Type                             25976 non-null  object
4   Age                                        25976 non-null  int64
5   Type of Travel                           25976 non-null  object
6   Class                                     25976 non-null  object
7   Flight Distance                          25976 non-null  int64
8   Inflight wifi service                    25976 non-null  int64
9   Departure/Arrival time convenient        25976 non-null  int64
10  Ease of Online booking                   25976 non-null  int64
11  Gate location                            25976 non-null  int64
12  Food and drink                           25976 non-null  int64
13  Online boarding                          25976 non-null  int64
14  Seat comfort                             25976 non-null  int64
15  Inflight entertainment                   25976 non-null  int64
16  On-board service                         25976 non-null  int64
17  Leg room service                         25976 non-null  int64
18  Baggage handling                         25976 non-null  int64
19  Checkin service                          25976 non-null  int64
20  Inflight service                         25976 non-null  int64
21  Cleanliness                             25976 non-null  int64
22  Departure Delay in Minutes               25976 non-null  int64
23  Arrival Delay in Minutes                 25893 non-null  float64
24  satisfaction                             25976 non-null  object
dtypes: float64(1), int64(19), object(5)
memory usage: 5.0+ MB

```

```
In [6]: df.describe()
```

Out[6]:

	Unnamed: 0	id	Age	Flight Distance
<b>count</b>	25976.000000	25976.000000	25976.000000	25976.000000
<b>mean</b>	12987.500000	65005.657992	39.620958	1193.788459
<b>std</b>	7498.769632	37611.526647	15.135685	998.683999
<b>min</b>	0.000000	17.000000	7.000000	31.000000
<b>25%</b>	6493.750000	32170.500000	27.000000	414.000000
<b>50%</b>	12987.500000	65319.500000	40.000000	849.000000
<b>75%</b>	19481.250000	97584.250000	51.000000	1744.000000
<b>max</b>	25975.000000	129877.000000	85.000000	4983.000000



In [7]: `df.isnull().sum()`

```

Out[7]: Unnamed: 0      0
        id             0
        Gender         0
        Customer Type  0
        Age            0
        Type of Travel 0
        Class          0
        Flight Distance 0
        Inflight wifi service 0
        Departure/Arrival time convenient 0
        Ease of Online booking 0
        Gate location  0
        Food and drink 0
        Online boarding 0
        Seat comfort   0
        Inflight entertainment 0
        On-board service 0
        Leg room service 0
        Baggage handling 0
        Checkin service 0
        Inflight service 0
        Cleanliness    0
        Departure Delay in Minutes 0
        Arrival Delay in Minutes 83
        satisfaction    0
        dtype: int64

```

```

In [8]: df["Arrival Delay in Minutes"].fillna(df["Arrival Delay in Minutes"])

```

```

In [9]: df.isnull().sum()

```

```
Out[9]: Unnamed: 0      0
        id             0
        Gender          0
        Customer Type   0
        Age             0
        Type of Travel   0
        Class           0
        Flight Distance  0
        Inflight wifi service 0
        Departure/Arrival time convenient 0
        Ease of Online booking 0
        Gate location    0
        Food and drink    0
        Online boarding   0
        Seat comfort      0
        Inflight entertainment 0
        On-board service  0
        Leg room service  0
        Baggage handling  0
        Checkin service   0
        Inflight service  0
        Cleanliness       0
        Departure Delay in Minutes 0
        Arrival Delay in Minutes 83
        satisfaction      0
        dtype: int64
```

```
In [10]: df.columns
```

```
Out[10]: Index(['Unnamed: 0', 'id', 'Gender', 'Customer Type', 'Age', 'Type of Travel', 'Class', 'Flight Distance', 'Inflight wifi service', 'Departure/Arrival time convenient', 'Ease of Online booking', 'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort', 'Inflight entertainment', 'On-board service', 'Leg room service', 'Baggage handling', 'Checkin service', 'Inflight service', 'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes', 'satisfaction'], dtype='object')
```

```
In [11]: df=df.drop(["Unnamed: 0","id"],axis=1)
```

```
In [12]: df
```

Out[12]:

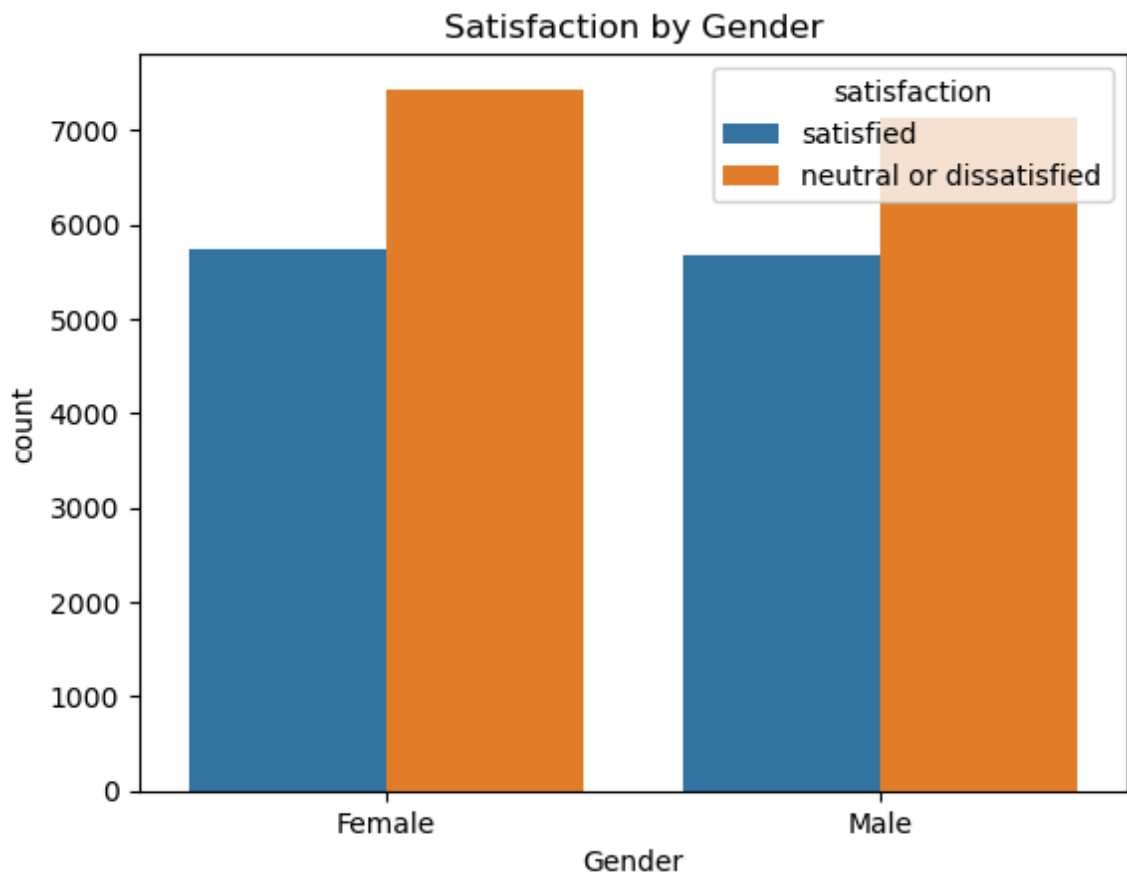
	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Infl ser
0	Female	Loyal Customer	52	Business travel	Eco	160	
1	Female	Loyal Customer	36	Business travel	Business	2863	
2	Male	disloyal Customer	20	Business travel	Eco	192	
3	Male	Loyal Customer	44	Business travel	Business	3377	
4	Female	Loyal Customer	49	Business travel	Eco	1182	
...	...	...	...	...	...	...	
25971	Male	disloyal Customer	34	Business travel	Business	526	
25972	Male	Loyal Customer	23	Business travel	Business	646	
25973	Female	Loyal Customer	17	Personal Travel	Eco	828	
25974	Male	Loyal Customer	14	Business travel	Business	1127	
25975	Female	Loyal Customer	42	Personal Travel	Eco	264	

25976 rows × 23 columns

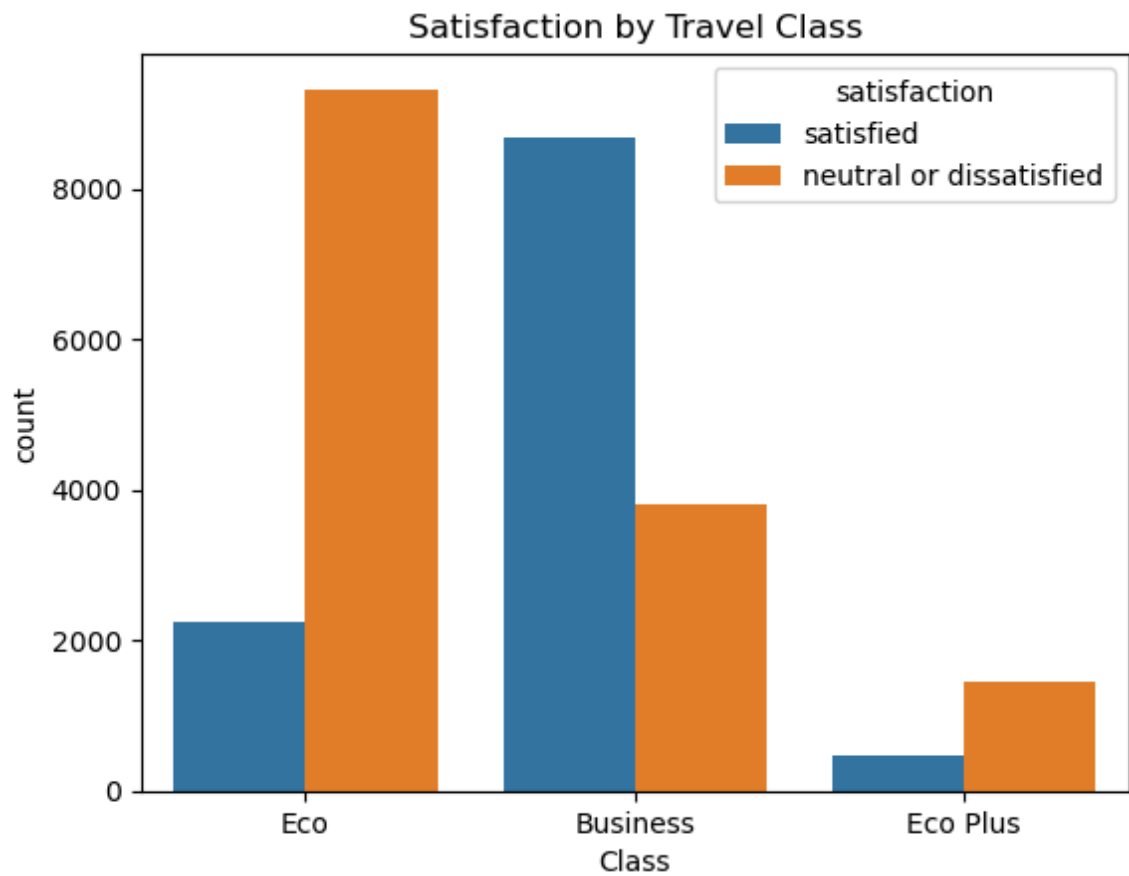


```
In [13]: #2. Gender vs Satisfaction
sns.countplot(data=df, x='Gender', hue='satisfaction')
plt.title("Satisfaction by Gender")
plt.show()
```

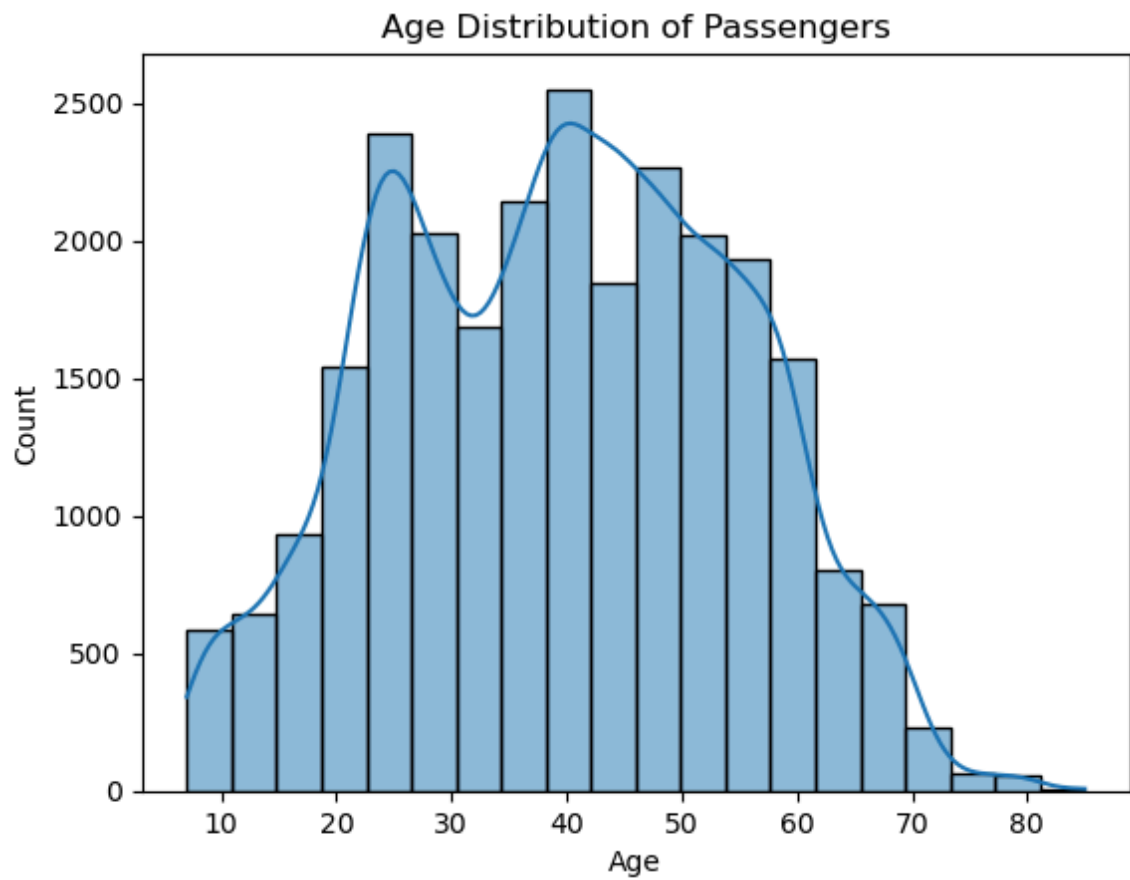




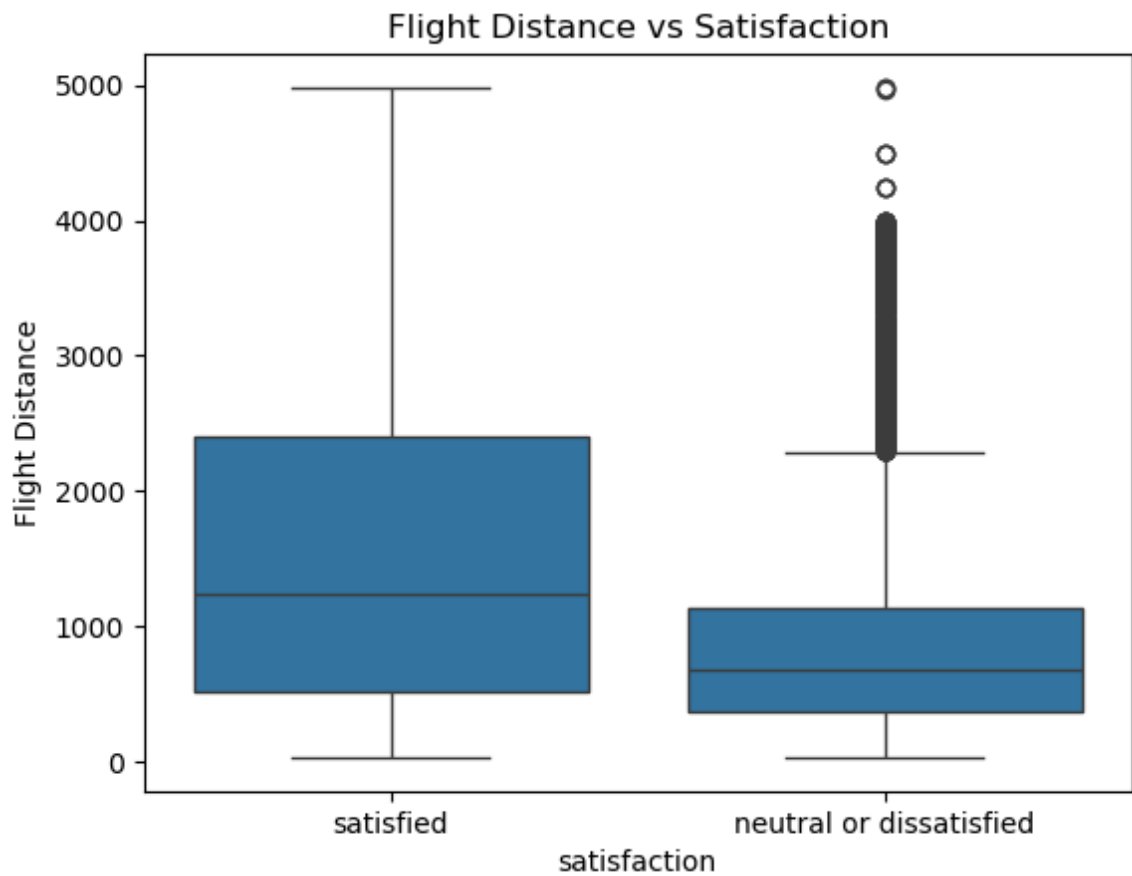
```
In [14]: # Class vs Satisfaction
sns.countplot(data=df, x='Class', hue='satisfaction')
plt.title("Satisfaction by Travel Class")
plt.show()
```



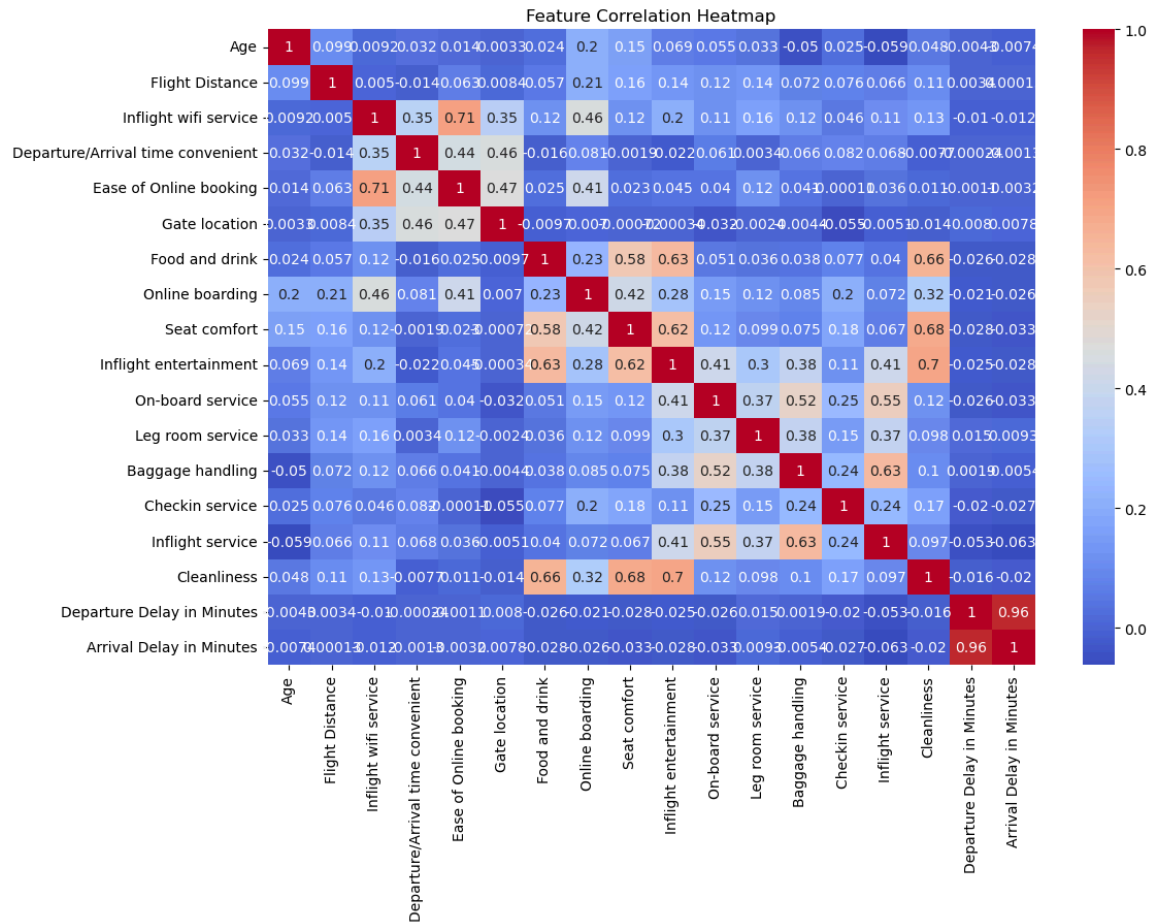
```
In [15]: #Age Distribution
sns.histplot(data=df, x='Age', bins=20, kde=True)
plt.title("Age Distribution of Passengers")
plt.show()
```



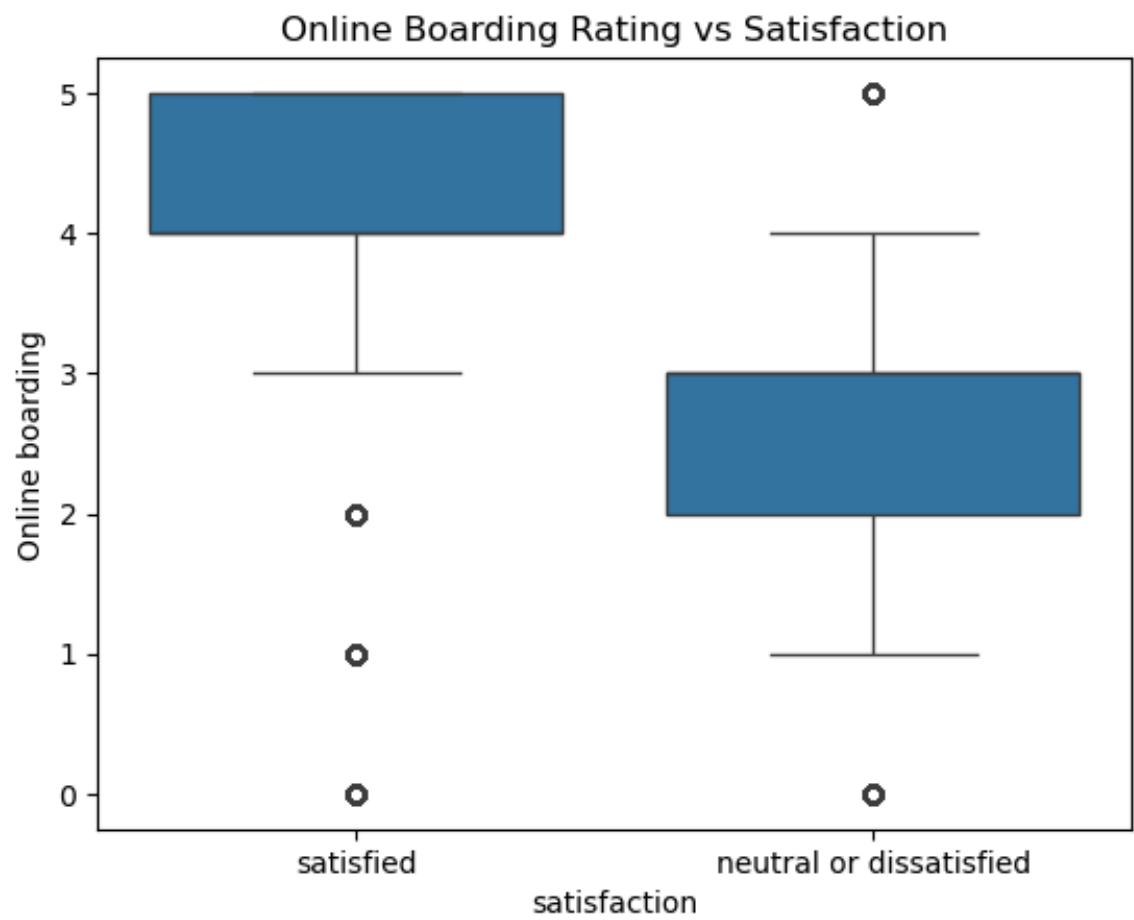
```
In [16]: #Flight Distance vs Satisfaction (Boxplot)
sns.boxplot(data=df, x='satisfaction', y='Flight Distance')
plt.title("Flight Distance vs Satisfaction")
plt.show()
```



```
In [17]: # Heatmap of Feature Correlation
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Feature Correlation Heatmap")
plt.show()
```



```
In [18]: # Online Boarding vs Satisfaction
sns.boxplot(x='satisfaction', y='Online boarding', data=df)
plt.title("Online Boarding Rating vs Satisfaction")
plt.show()
```



```
In [19]: df.isnull().sum()
```

```
Out[19]: Gender      0
         Customer Type 0
         Age          0
         Type of Travel 0
         Class        0
         Flight Distance 0
         Inflight wifi service 0
         Departure/Arrival time convenient 0
         Ease of Online booking 0
         Gate location 0
         Food and drink 0
         Online boarding 0
         Seat comfort 0
         Inflight entertainment 0
         On-board service 0
         Leg room service 0
         Baggage handling 0
         Checkin service 0
         Inflight service 0
         Cleanliness 0
         Departure Delay in Minutes 0
         Arrival Delay in Minutes 83
         satisfaction 0
         dtype: int64
```

```
In [21]: df.dropna(inplace=True)
```

```
In [22]: df.isnull().sum()
```

```
Out[22]: Gender                                0
         Customer Type                         0
         Age                                   0
         Type of Travel                        0
         Class                                 0
         Flight Distance                      0
         Inflight wifi service                0
         Departure/Arrival time convenient    0
         Ease of Online booking               0
         Gate location                        0
         Food and drink                       0
         Online boarding                      0
         Seat comfort                         0
         Inflight entertainment               0
         On-board service                     0
         Leg room service                     0
         Baggage handling                     0
         Checkin service                      0
         Inflight service                     0
         Cleanliness                          0
         Departure Delay in Minutes           0
         Arrival Delay in Minutes             0
         satisfaction                          0
         dtype: int64
```

```
In [20]: from sklearn.model_selection import train_test_split
         from sklearn.preprocessing import LabelEncoder, StandardScaler
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import accuracy_score, classification_re
```

```
In [23]: le=LabelEncoder()
```

```
In [25]: cat_c=df.select_dtypes("object")
         cat_c
```



Out[25]:

	Gender	Customer Type	Type of Travel	Class	satisfaction
0	Female	Loyal Customer	Business travel	Eco	satisfied
1	Female	Loyal Customer	Business travel	Business	satisfied
2	Male	disloyal Customer	Business travel	Eco	neutral or dissatisfied
3	Male	Loyal Customer	Business travel	Business	satisfied
4	Female	Loyal Customer	Business travel	Eco	satisfied
...	...	...	...	...	...
25971	Male	disloyal Customer	Business travel	Business	neutral or dissatisfied
25972	Male	Loyal Customer	Business travel	Business	satisfied
25973	Female	Loyal Customer	Personal Travel	Eco	neutral or dissatisfied
25974	Male	Loyal Customer	Business travel	Business	satisfied
25975	Female	Loyal Customer	Personal Travel	Eco	neutral or dissatisfied

25893 rows × 5 columns

```
In [27]: for i in cat_c:  
         cat_c[i]=le.fit_transform(cat_c[i])
```

```
In [29]: cat_c
```

Out[29]:

	Gender	Customer Type	Type of Travel	Class	satisfaction
<b>0</b>	0	0	0	1	1
<b>1</b>	0	0	0	0	1
<b>2</b>	1	1	0	1	0
<b>3</b>	1	0	0	0	1
<b>4</b>	0	0	0	1	1
...	...	...	...	...	...
<b>25971</b>	1	1	0	0	0
<b>25972</b>	1	0	0	0	1
<b>25973</b>	0	0	1	1	0
<b>25974</b>	1	0	0	0	1
<b>25975</b>	0	0	1	1	0

25893 rows × 5 columns

```
In [30]: num = df.select_dtypes(["int64","float64"])
```

```
In [31]: num
```

Out[31]:

	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	G locat
<b>0</b>	52	160	5	4	3	
<b>1</b>	36	2863	1	1	3	
<b>2</b>	20	192	2	0	2	
<b>3</b>	44	3377	0	0	0	
<b>4</b>	49	1182	2	3	4	
...	...	...	...	...	...	
<b>25971</b>	34	526	3	3	3	
<b>25972</b>	23	646	4	4	4	
<b>25973</b>	17	828	2	5	1	
<b>25974</b>	14	1127	3	3	3	
<b>25975</b>	42	264	2	5	2	


25893 rows × 18 columns



In [33]: num.describe()

Out[33]:

	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient
count	25893.000000	25893.000000	25893.000000	25893.000000
mean	39.621983	1193.753254	2.723709	3.046420
std	15.134224	998.626779	1.334711	1.532970
min	7.000000	31.000000	0.000000	0.000000
25%	27.000000	414.000000	2.000000	2.000000
50%	40.000000	849.000000	3.000000	3.000000
75%	51.000000	1744.000000	4.000000	4.000000
max	85.000000	4983.000000	5.000000	5.000000



In [34]: `df_final=pd.concat([cat_c,num],axis=1)`  
`df_final`

Out[34]:

	Gender	Customer Type	Type of Travel	Class	satisfaction	Age	Flight Distance
0	0	0	0	1	1	52	...
1	0	0	0	0	1	36	28
2	1	1	0	1	0	20	...
3	1	0	0	0	1	44	33
4	0	0	0	1	1	49	17
...	...	...	...	...	...	...	...
25971	1	1	0	0	0	34	...
25972	1	0	0	0	1	23	...
25973	0	0	1	1	0	17	...
25974	1	0	0	0	1	14	17
25975	0	0	1	1	0	42	...

25893 rows × 23 columns



```
In [41]: x=df_final.drop(columns=["satisfaction"])  
y=df_final["satisfaction"]
```

```
In [39]: x
```

Out[39]:

	Gender	Customer Type	Type of Travel	Class	Age	Flight Distance	Inflight wifi service
<b>0</b>	0	0	0	1	52	160	5
<b>1</b>	0	0	0	0	36	2863	1
<b>2</b>	1	1	0	1	20	192	2
<b>3</b>	1	0	0	0	44	3377	0
<b>4</b>	0	0	0	1	49	1182	2
<b>...</b>	...	...	...	...	...	...	...
<b>25971</b>	1	1	0	0	34	526	3
<b>25972</b>	1	0	0	0	23	646	4
<b>25973</b>	0	0	1	1	17	828	2
<b>25974</b>	1	0	0	0	14	1127	3
<b>25975</b>	0	0	1	1	42	264	2

25893 rows × 22 columns



In [42]:

y

Out[42]:

```
0      1
1      1
2      0
3      1
4      1
..
25971  0
25972  1
25973  0
25974  1
25975  0
```

Name: satisfaction, Length: 25893, dtype: int32

```
In [44]: from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,train_size=0.
```

```
In [45]: from sklearn.linear_model import LogisticRegression
```

```
In [47]: log=LogisticRegression(max_iter=1000)
log.fit(xtrain,ytrain)
pred=log.predict(xtest)
```

C:\Users\Mohan M\anaconda3\Lib\site-packages\sklearn\linear\_model\\_logistic.py:469: 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](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
```

```
In [48]: log.score(xtrain,ytrain)
```

```
Out[48]: 0.8592
```

```
In [49]: from sklearn.metrics import accuracy_score,classification_re
```

```
In [50]: accuracy_score(ytest,pred)
```

```
Out[50]: 0.865859938208033
```

```
In [51]: print(classification_report(ytest,pred))
```

	precision	recall	f1-score	support
0	0.88	0.89	0.88	4397
1	0.85	0.84	0.84	3371
accuracy			0.87	7768
macro avg	0.86	0.86	0.86	7768
weighted avg	0.87	0.87	0.87	7768

```
In [64]: print("Accuracy Score for test: ",accuracy_score(ytest,pred))
print("Accuracy Score for Trian: ",accuracy_score(ytest,pred))
```

```
Accuracy Score for test: 0.865859938208033
Accuracy Score for Trian: 0.865859938208033
```

```
In [52]: from sklearn.metrics import roc_curve,roc_auc_score
```

```
In [53]: list(ytest)[:20]
```

```
Out[53]: [0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1]
```

```
In [54]: pred
```

```
Out[54]: array([0, 1, 0, ..., 1, 1, 0])
```

```
In [55]: pred_prob=log.predict_proba(xtest)[:,-1]
pred_prob
```

```
Out[55]: array([0.18780659, 0.52690782, 0.32146452, ..., 0.97798206,
0.85431265,
0.25347751])
```

```
In [56]: fbr,tpr,threshold=roc_curve(ytest,pred_prob)
```

```
In [57]: auc_score=roc_auc_score(ytest,pred_prob)
```

```
In [58]: auc_score
```

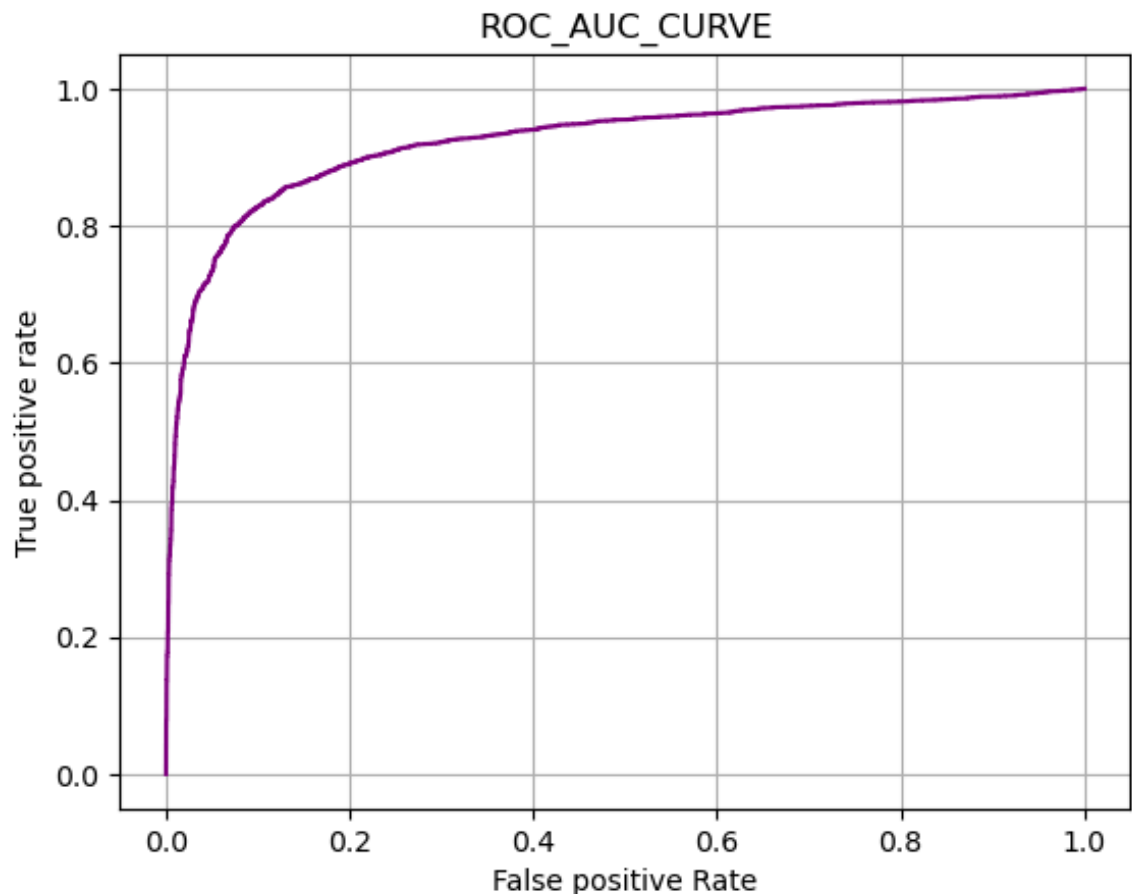
```
Out[58]: 0.9228721586621552
```



```
In [59]: print(threshold)
print(fbr)
print(tpr)
```

```
[          inf  9.94620288e-01  9.81971848e-01 ...  1.88536790e
-03
 1.86798211e-03  2.96115676e-04]
[0.          0.          0.          ...  0.99840801  0.99840801
 1.          ]
[0.00000000e+00  2.96647879e-04  1.77988727e-02 ...  9.99703352e
-01
 1.00000000e+00  1.00000000e+00]
```

```
In [62]: plt.plot(fbr,tpr,label=f"ROC-CURVE(AUC={auc_score:.2f})",col
plt.xlabel("False positive Rate")
plt.ylabel("True positive rate")
plt.title("ROC_AUC_CURVE")
plt.grid(True)
plt.show()
```



In [ ]: