In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:

```
df=pd.read_csv(r"C:\Users\DELL\Downloads\ML Project - Decision Tree Airline Customer Satisfaction U1695546898
```

In [4]:

```
df
```

Out[4]:

	Unnamed: 0.1	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	...	Inflight entertainment	On-board service	Leg room servic
0	0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	...	5	4	3
1	1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	...	1	1	5
2	2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	...	5	4	3
3	3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	...	2	2	5
4	4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	...	3	3	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9995	9995	124365	Male	Loyal Customer	50	Business travel	Business	3599	3	...	5	5	5
9996	9996	9996	22044	Male	Loyal Customer	38	Business travel	Business	3873	5	...	4	4	4
9997	9997	9997	14057	Female	Loyal Customer	39	Business travel	Business	319	4	...	4	4	4
9998	9998	9998	113848	Male	Loyal Customer	52	Business travel	Business	1363	5	...	4	4	3
9999	9999	9999	1865	Female	Loyal Customer	41	Business travel	Business	3938	4	...	5	5	4

10000 rows × 26 columns

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

Out[5]:

	Unnamed: 0.1	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	...	Inflight entertainment	On-board service	Leg room service
0	0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	...	5	4	3
1	1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	...	1	1	5
2	2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	...	5	4	3
3	3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	...	2	2	5
4	4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	...	3	3	4

5 rows × 26 columns

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Unnamed: 0.1', '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 [7]:
```
df.describe()
```

Out[7]:

	Unnamed: 0.1	Unnamed: 0	id	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Ga locatic
count	10000.00000	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.0000
mean	4999.50000	4999.50000	64950.978500	39.303700	1198.823800	2.731500	3.038900	2.753600	2.968
std	2886.89568	2886.89568	37420.905082	15.116478	1001.774138	1.328226	1.528077	1.403241	1.284
min	0.00000	0.00000	8.000000	7.000000	31.000000	0.000000	0.000000	0.000000	1.0000
25%	2499.75000	2499.75000	32330.500000	27.000000	413.000000	2.000000	2.000000	2.000000	2.0000
50%	4999.50000	4999.50000	64941.000000	40.000000	852.500000	3.000000	3.000000	3.000000	3.0000
75%	7499.25000	7499.25000	97579.750000	51.000000	1747.000000	4.000000	4.000000	4.000000	4.0000
max	9999.00000	9999.00000	129863.000000	85.000000	4983.000000	5.000000	5.000000	5.000000	5.0000

8 rows × 21 columns

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

In [12]:

```
df
```

Out[12]:

	Unnamed: 0.1	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...	5	4
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	...	1	1
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	...	5	4
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	...	2	2
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	...	3	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9995	124365	Male	Loyal Customer	50	Business travel	Business	3599	3	3	...	5	5
9996	9996	22044	Male	Loyal Customer	38	Business travel	Business	3873	5	5	...	4	4
9997	9997	14057	Female	Loyal Customer	39	Business travel	Business	319	4	4	...	4	4
9998	9998	113848	Male	Loyal Customer	52	Business travel	Business	1363	5	5	...	4	4
9999	9999	1865	Female	Loyal Customer	41	Business travel	Business	3938	4	4	...	5	5

10000 rows × 25 columns

In [13]:

```
df.columns=[each.replace(" ","_")for each in df.columns]
```

In [14]:
```
df.columns
```

Out[14]:
```
Index(['Unnamed:_0.1', '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')
```

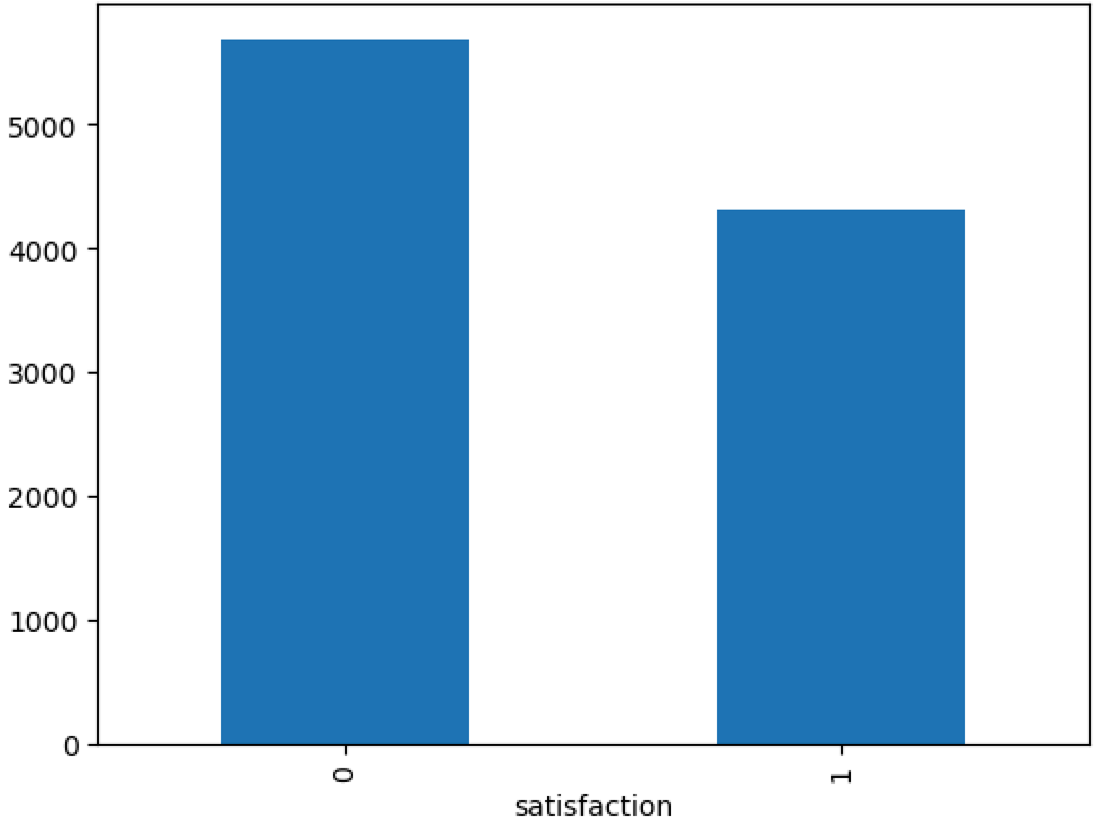
In [18]:
```
df["satisfaction"]=[1 if each=="satisfied" else 0 for each in df.satisfaction]
```

In [19]:
```
df["satisfaction"].value_counts()
```

Out[19]:
```
satisfaction
0    5689
1    4311
Name: count, dtype: int64
```

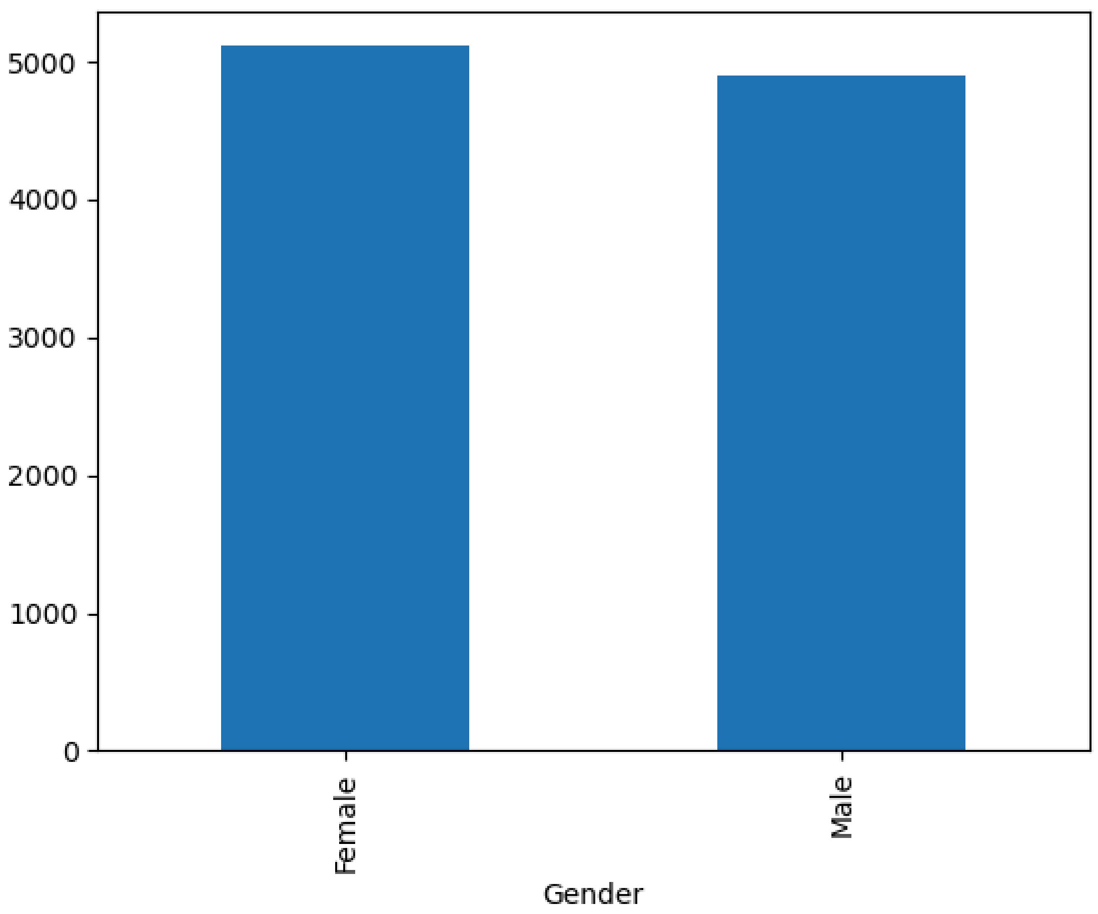
In [20]:
```
df["satisfaction"].value_counts().plot(kind="bar")
```

Out[20]:
```
<Axes: xlabel='satisfaction'>
```



In [22]:
```
df["Gender"].value_counts().plot(kind="bar")
```

Out[22]:
```
<Axes: xlabel='Gender'>
```



In [25]:
```
#gender vs Satisfaction
df[["Gender","satisfaction"]].groupby(['Gender'],as_index=False).mean().sort_values(by="satisfaction",ascendi
```

Out[25]:

	Gender	satisfaction
1	Male	0.438062
0	Female	0.424432

In [26]:

```python
#age vs satisfacetion
df[["Age","satisfaction"]].groupby(['Age'],as_index=False).mean().sort_values(by="satisfaction",ascending=Fal
```

Out[26]:

	Age	satisfaction
71	79	0.666667
39	46	0.643192
34	41	0.632743
38	45	0.608466
66	73	0.600000
...	...	...
7	14	0.076923
0	7	0.071429
59	66	0.050000
69	76	0.000000
73	85	0.000000

74 rows × 2 columns

In [27]:

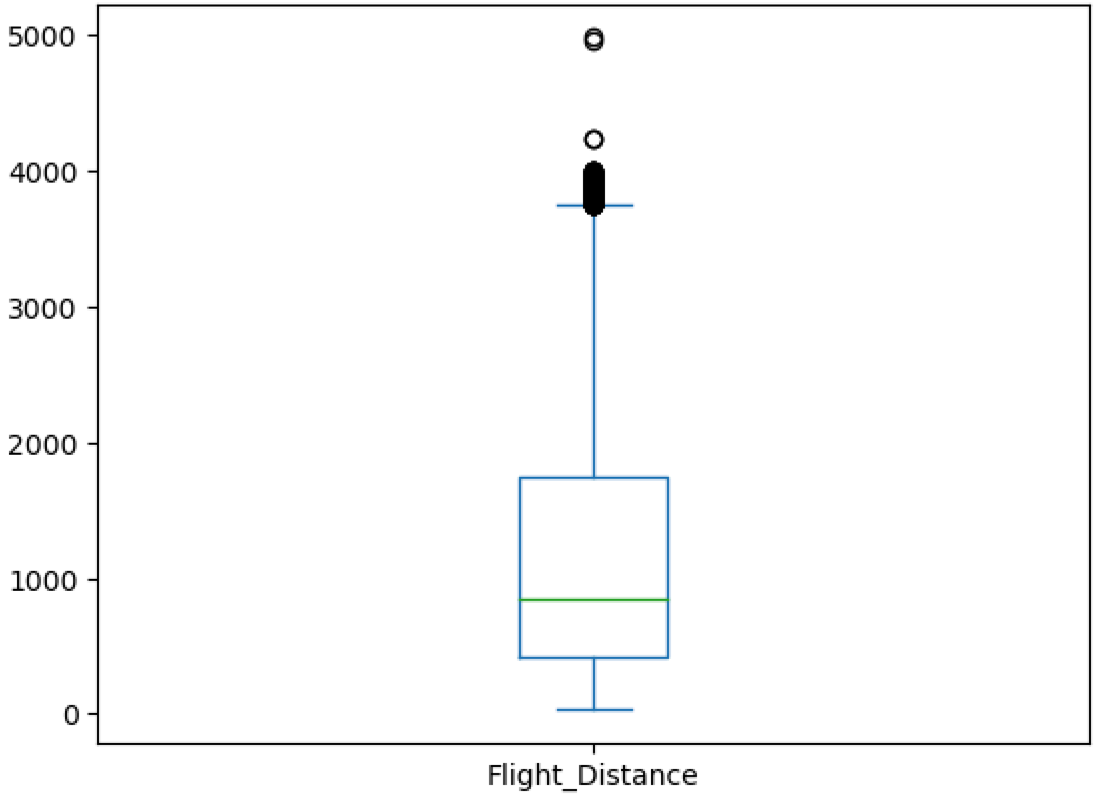
```python
#food and drink vs satisfacetion
df[["Food_and_drink","satisfaction"]].groupby(['Food_and_drink'],as_index=False).mean().sort_values(by="sati
```

Out[27]:

	Food_and_drink	satisfaction
0	0	0.625000
5	5	0.566176
4	4	0.515050
3	3	0.404990
2	2	0.362305
1	1	0.198142

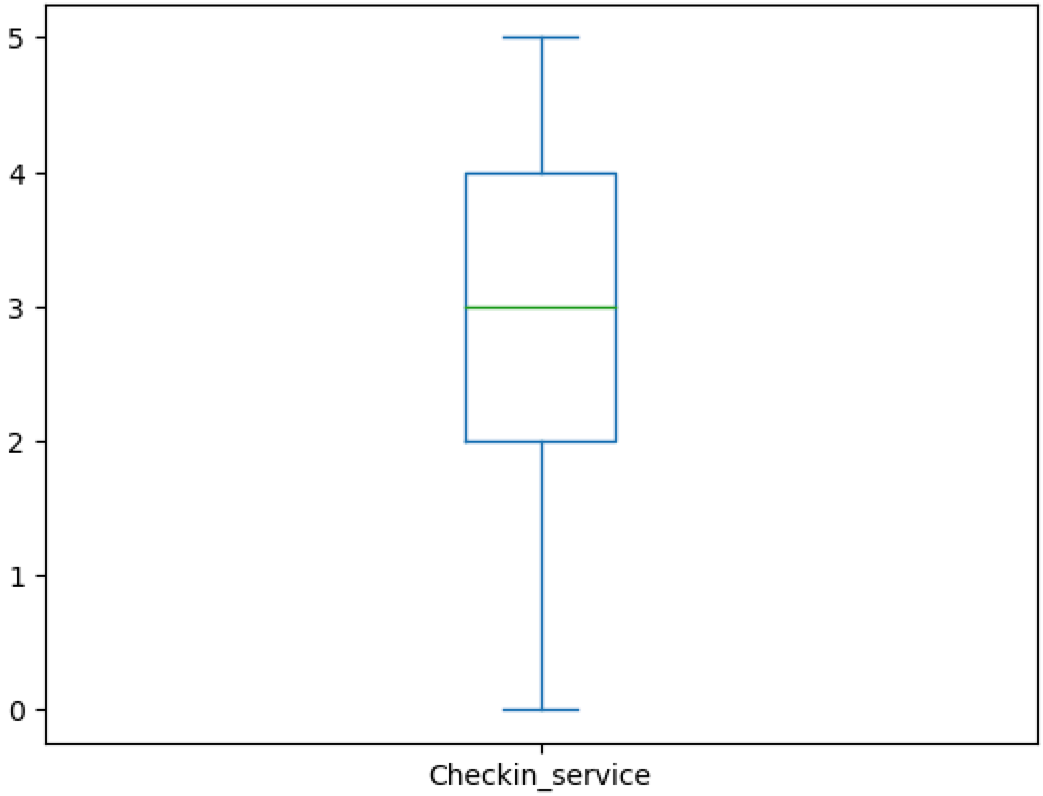
In [29]:
```
df["Flight_Distance"].plot(kind="box")
```

Out[29]:
```
<Axes: >
```



In [30]:
```
df["Checkin_service"].plot(kind="box")
```

Out[30]:
```
<Axes: >
```



In [32]:
```
df.isnull().sum()
```

Out[32]:
```
Unnamed:_0.1                          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             26
satisfaction                          0
dtype: int64
```

In [33]:
```
df=df.dropna()
```

In [34]:
```
df.isnull().sum()
```

Out[34]:
```
Unnamed:_0.1                         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             0
satisfaction                         0
dtype: int64
```

In [35]:
```
df["Flight_Distance"].unique()
```

Out[35]:
```
array([ 460,  235, 1142, ..., 3033, 1319, 1443], dtype=int64)
```

part 2

In [36]:
```
from sklearn.preprocessing import LabelEncoder
```

In [37]:
```
enc=LabelEncoder()
```

In [88]:
```
import warnings
warnings.filterwarnings ("ignore")
```

In [89]:
```
df['Gender'] = enc.fit_transform(df['Gender'])
df["Customer_Type"]= enc.fit_transform(df["Customer_Type"])
df["Type_of_Travel"]=enc.fit_transform(df["Type_of_Travel"])
df["Class"]=enc.fit_transform(df["Class"])
```

In [44]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9974 entries, 0 to 9999
Data columns (total 25 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Unnamed:_0.1                       9974 non-null   int64  
 1   id                                 9974 non-null   int64  
 2   Gender                             9974 non-null   int64  
 3   Customer_Type                      9974 non-null   int64  
 4   Age                                9974 non-null   int64  
 5   Type_of_Travel                     9974 non-null   int64  
 6   Class                              9974 non-null   int32  
 7   Flight_Distance                    9974 non-null   int64  
 8   Inflight_wifi_service              9974 non-null   int64  
 9   Departure/Arrival_time_convenient  9974 non-null   int64  
 10  Ease_of_Online_booking             9974 non-null   int64  
 11  Gate_location                      9974 non-null   int64  
 12  Food_and_drink                     9974 non-null   int64  
 13  Online_boarding                    9974 non-null   int64  
 14  Seat_comfort                       9974 non-null   int64  
 15  Inflight_entertainment             9974 non-null   int64  
 16  On-board_service                   9974 non-null   int64  
 17  Leg_room_service                   9974 non-null   int64  
 18  Baggage_handling                   9974 non-null   int64  
 19  Checkin_service                    9974 non-null   int64  
 20  Inflight_service                   9974 non-null   int64  
 21  Cleanliness                        9974 non-null   int64  
 22  Departure_Delay_in_Minutes         9974 non-null   int64  
 23  Arrival_Delay_in_Minutes           9974 non-null   float64
 24  satisfaction                       9974 non-null   int64  
dtypes: float64(1), int32(1), int64(23)
memory usage: 1.9 MB
```

In [48]:
```
df = df.drop(['id','Unnamed:_0.1'],axis=1)
```

In [49]:
```
df.head()
```

Out[49]:

ce	Inflight_wifi_service	Departure/Arrival_time_convenient	Ease_of_Online_booking	Gate_location	...	Inflight_entertainment	On board_servic
60	3	4	3	1	...	5	
35	3	2	3	3	...	1	
42	2	2	2	2	...	5	
62	2	5	5	5	...	2	
14	3	3	3	3	...	3	

In [50]:
```
X = df.drop('satisfaction',axis=1)
```

In [51]:
```
y = df['satisfaction']
```

In [53]:

```
X
```

Out[53]:

	Gender	Customer_Type	Age	Type_of_Travel	Class	Flight_Distance	Inflight_wifi_service	Departure/Arrival_time_convenient	Eas
0	1	0	13	1	2	460	3	4	
1	1	1	25	0	0	235	3	2	
2	0	0	26	0	0	1142	2	2	
3	0	0	25	0	0	562	2	5	
4	1	0	61	0	0	214	3	3	
...	...	...	...	...	...	...	...	...	
9995	1	0	50	0	0	3599	3	3	
9996	1	0	38	0	0	3873	5	5	
9997	0	0	39	0	0	319	4	4	
9998	1	0	52	0	0	1363	5	5	
9999	0	0	41	0	0	3938	4	4	

9974 rows × 22 columns

In [54]:

```
y
```

Out[54]:

```
0       0
1       0
2       1
3       0
4       1
       ..
9995    1
9996    1
9997    1
9998    1
9999    1
Name: satisfaction, Length: 9974, dtype: int64
```

In [55]:
```
X.shape
```

Out[55]:
```
(9974, 22)
```

In [56]:
```
from sklearn.preprocessing import StandardScaler
```

In [57]:
```
scaler= StandardScaler()
```

In [59]:
```
X_scal=scaler.fit_transform(X)
```

In [60]:
```
from sklearn.model_selection import train_test_split
```

In [61]:
```
x_train,x_test,y_train,y_test= train_test_split(X_scal,y,test_size=0.2)
```

In [62]:
```
x_train.shape
```

Out[62]:
```
(7979, 22)
```

In [63]:
```
x_test.shape
```

Out[63]:
```
(1995, 22)
```

#treemodel

In [65]:
```
from sklearn.tree import DecisionTreeClassifier
import sklearn.tree as tree
```

In [66]:
```
model=DecisionTreeClassifier()
```

```
In [67]: model.fit(x_train,y_train)
```

```
Out[67]: DecisionTreeClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [68]: model.score(x_train,y_train)
```

```
Out[68]: 1.0
```

```
In [69]: model.score(x_test,y_test)
```

```
Out[69]: 0.9223057644110275
```

```
In [70]: from sklearn.metrics import accuracy_score,confusion_matrix,f1_score,classification_report,recall_score,prec
```

```
In [79]: def run_model(model, X_train, y_train, X_test, y_test):
             model.fit(X_train,y_train.ravel())
             y_pred = model.predict(X_test)
             import seaborn as snas
```

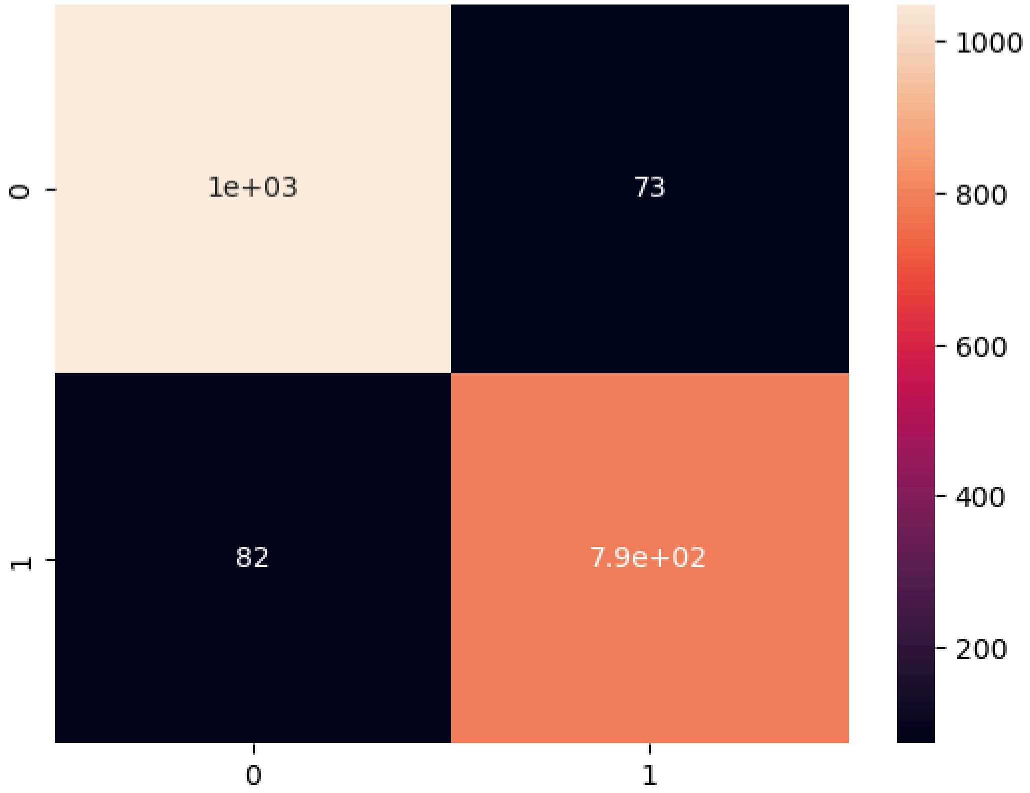
```
In [80]: confusion_matrix(y_test,model.predict(x_test))
```

```
Out[80]: array([[1049,   73],
                [  82,  791]], dtype=int64)
```

```
In [82]:  import seaborn as sns
```

In [83]:
```
sns.heatmap(confusion_matrix(y_test,model.predict(x_test)),annot=True)
```

Out[83]:
```
<Axes: >
```



In [84]:
```
print(classification_report(y_test,model.predict(x_test)))
```

```
              precision    recall  f1-score   support

           0       0.93      0.93      0.93      1122
           1       0.92      0.91      0.91       873

    accuracy                           0.92      1995
   macro avg       0.92      0.92      0.92      1995
weighted avg       0.92      0.92      0.92      1995
```

In [ ]: