Project on Airline customer satisfaction prediction using Machine Learning

- **Aim:-**To create a Data science Project, where we will be predicting the Airline customer satisfaction. The change in airline passengers' behaviour following the pandemic crisis, travel restrictions, the ensuing economic crisis, market liberalization, high technology, and reorganization has resulted in airline services.

  Prediction Airline customer satisfaction with help of :-

  Age, Ease of Online booking, Food and drink, Departure/Arrival time convenient, Check-in service, On-board service etc. are some facility provided by Airline.
- Steps to be taken in the project is sub-divided into the following sections. These are:
  - Importing the libraries such as 'numpy', 'pandas','sklearn. model' etc.
  - Loading Dataset as a CSV file for training & testing the models.
  - Splitting the data set into independent & dependent sets.
  - Checking if still any null values or any other data types other than float and integers are present into the dataset or not.
  - Importing the train_test_split model from sklearn.model for splitting data into train & test sets.
  - Applying the different kinds of ML Algorithms .which gives Best accuracy of model.
  - Also checking with new data set for predicting the values.
- Steps of creating ML model:-
- Importing numpy as np & pandas as pd for loading and reading the data-set & using matplotlib.pyplot and Seaborn for visualization of data.

```
[1]
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as snr
```

- Loading the csv-dataset in the variable name 'train_data' Then viewing the data with train_data.head()

```
[2] train_data=pd.read_csv('/content/train_dataset (1).csv')
    train_data.head()
```

	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding	Seat comfort	Inflight entertainment	...	Departure Delay in Minutes	Arrival Delay in Minutes	Gender_Female	Gende
0	35	971	3	4	5	4	2	3	3	2	...	373	358.0	0	
1	32	1092	0	0	0	3	1	0	1	1	...	0	0.0	0	
2	46	2915	0	5	0	5	3	4	5	1	...	0	0.0	1	
3	56	2556	4	4	4	4	4	4	4	3	...	19	18.0	0	
4	54	468	1	4	1	4	4	1	4	4	...	0	0.0	0	

5 rows × 26 columns

```
[2] train_data=pd.read_csv('/content/train_dataset (1).csv')
    train_data.head()
```

Seat comfort	Inflight entertainment	...	Departure Delay in Minutes	Arrival Delay in Minutes	Gender_Female	Gender_Male	Type of Travel_Business travel	Type of Travel_Personal Travel	Class_Business	Class_Eco	Class_Eco Plus	satisfaction
3	2	...	373	358.0	0	1	1	0	0	1	0	neutral or dissatisfied
1	1	...	0	0.0	0	1	1	0	0	1	0	satisfied
5	1	...	0	0.0	1	0	1	0	1	0	0	satisfied
4	3	...	19	18.0	0	1	1	0	1	0	0	satisfied
4	4	...	0	0.0	0	1	0	1	0	1	0	neutral or dissatisfied

- Checking the data such as number of columns, rows and type of data(float,integer) with help of train _data.info()

```
[4] train_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50761 entries, 0 to 50760
Data columns (total 26 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Age                               50761 non-null  int64
 1   Flight Distance                   50761 non-null  int64
 2   Inflight wifi service             50761 non-null  int64
 3   Departure/Arrival time convenient 50761 non-null  int64
 4   Ease of Online booking            50761 non-null  int64
 5   Gate location                     50761 non-null  int64
 6   Food and drink                    50761 non-null  int64
 7   Online boarding                   50761 non-null  int64
 8   Seat comfort                      50761 non-null  int64
 9   Inflight entertainment            50761 non-null  int64
 10  On-board service                  50761 non-null  int64
 11  Leg room service                  50761 non-null  int64
 12  Baggage handling                  50761 non-null  int64
 13  Checkin service                   50761 non-null  int64
 14  Inflight service                  50761 non-null  int64
 15  Cleanliness                       50761 non-null  int64
 16  Departure Delay in Minutes        50761 non-null  int64
 17  Arrival Delay in Minutes          50761 non-null  float64
 18  Gender_Female                     50761 non-null  int64
 19  Gender_Male                       50761 non-null  int64
 20  Type of Travel_Business travel    50761 non-null  int64
 21  Type of Travel_Personal Travel    50761 non-null  int64
 22  Class_Business                    50761 non-null  int64
 23  Class_Eco                         50761 non-null  int64
 24  Class_Eco Plus                    50761 non-null  int64
 25  satisfaction                      50761 non-null  object
dtypes: float64(1), int64(24), object(1)
```

We observe that the above data have integer, object and float.

```
[58] train_data.shape

     (50761, 26)
```

Train data have 50761 Rows and 26 columns

- Now checking data have Nan value or not.

```
[5] train_data.isnull().sum()
```

```
Age                                  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
Gender_Female                        0
Gender_Male                          0
Type of Travel_Business travel       0
Type of Travel_Personal Travel       0
Class_Business                       0
Class_Eco                            0
Class_Eco Plus                       0
satisfaction                         0
dtype: int64
```

We observe that the above data have not Nan value.

- Now,Main focus convert the categorical data into Numerical data with help of one hot encoding method.

```
[6] train_data['satisfaction']=train_data['satisfaction'].replace({'neutral or dissatisfied':0,'satisfied':1})
```

```
train_data
```

	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding	Seat comfort	Inflight entertainment	...	Departure Delay in Minutes	Arrival Delay in Minutes	Gender_Female	Gender_M
0	35	971	3	4	5	4	2	3	3	2	...	373	358.0	0	
1	32	1092	0	0	0	3	1	0	1	1	...	0	0.0	0	
2	46	2915	0	5	0	5	3	4	5	1	...	0	0.0	1	
3	56	2556	4	4	4	4	4	4	4	3	...	19	18.0	0	
4	54	468	1	4	1	4	4	1	4	4	...	0	0.0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
50756	41	3937	4	4	4	4	2	5	4	3	...	57	57.0	1	
50757	40	265	5	5	5	5	5	5	5	5	...	0	0.0	1	
50758	51	1750	1	1	1	1	4	3	4	5	...	22	24.0	0	
50759	18	296	3	3	3	3	4	3	4	4	...	0	0.0	1	
50760	24	581	3	5	3	5	4	3	4	4	...	53	58.0	0	

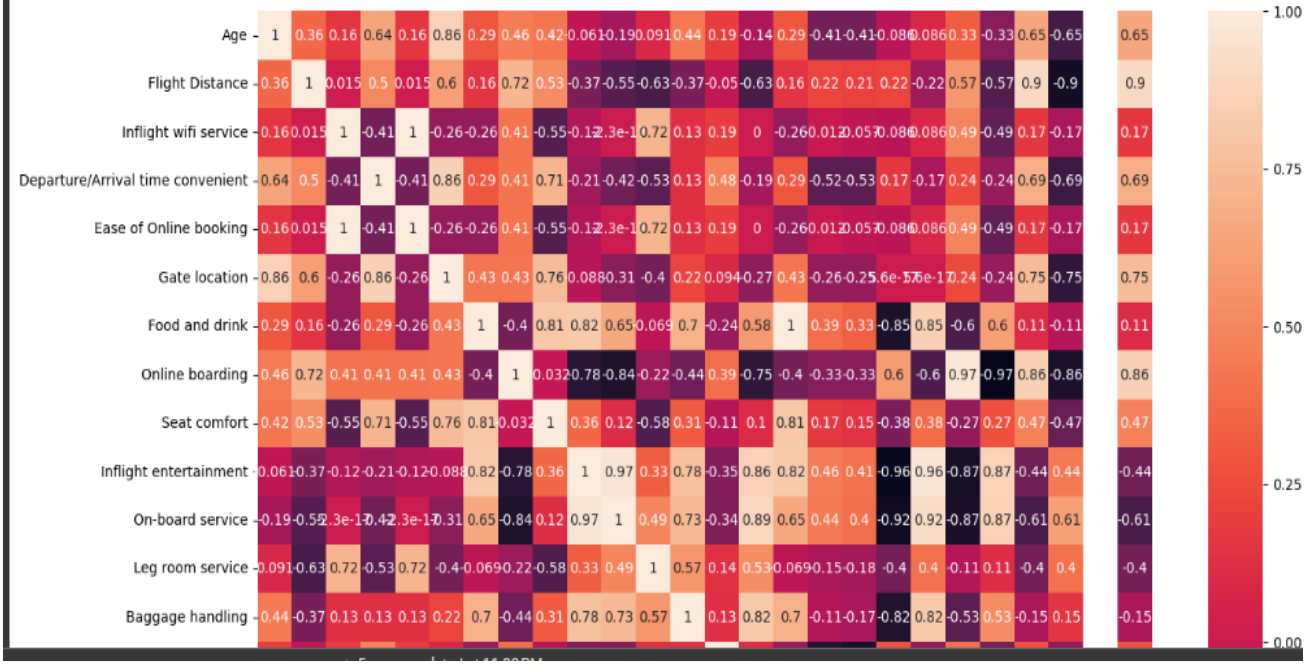
50761 rows × 26 columns

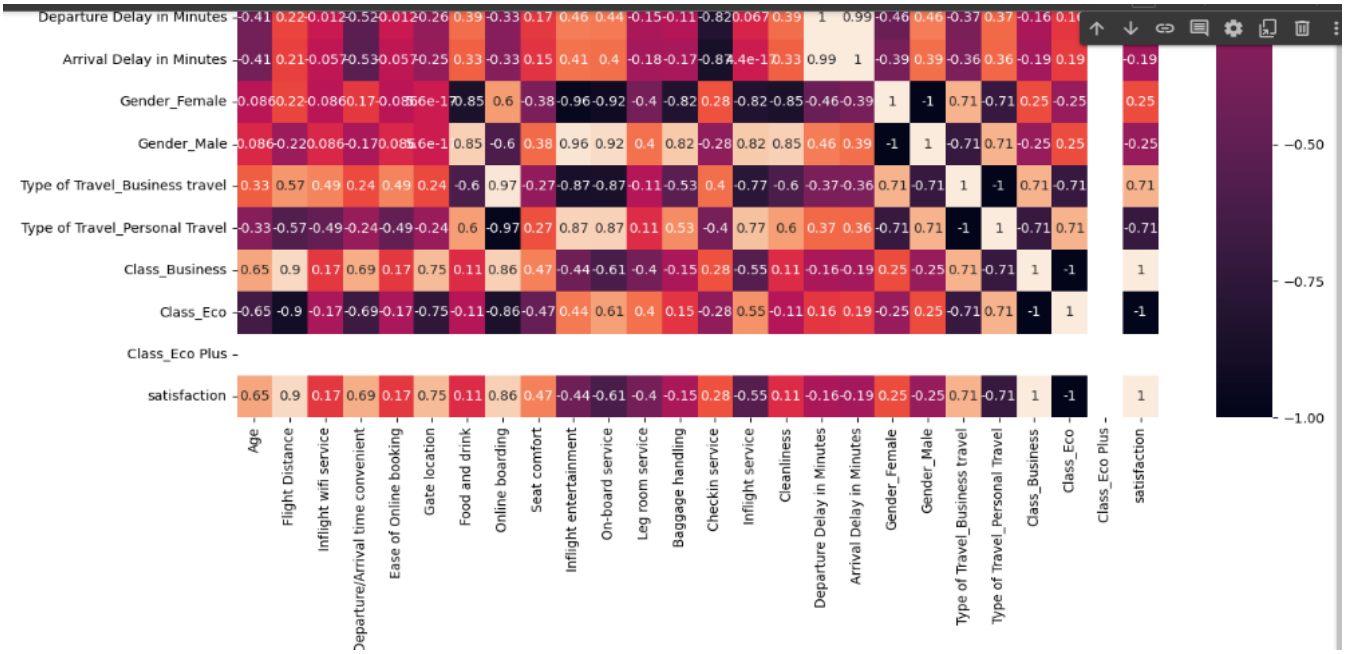
```
[8]  train_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50761 entries, 0 to 50760
Data columns (total 26 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Age                                50761 non-null  int64
 1   Flight Distance                    50761 non-null  int64
 2   Inflight wifi service              50761 non-null  int64
 3   Departure/Arrival time convenient  50761 non-null  int64
 4   Ease of Online booking             50761 non-null  int64
 5   Gate location                      50761 non-null  int64
 6   Food and drink                     50761 non-null  int64
 7   Online boarding                    50761 non-null  int64
 8   Seat comfort                       50761 non-null  int64
 9   Inflight entertainment             50761 non-null  int64
 10  On-board service                   50761 non-null  int64
 11  Leg room service                   50761 non-null  int64
 12  Baggage handling                   50761 non-null  int64
 13  Checkin service                    50761 non-null  int64
 14  Inflight service                   50761 non-null  int64
 15  Cleanliness                        50761 non-null  int64
 16  Departure Delay in Minutes         50761 non-null  int64
 17  Arrival Delay in Minutes           50761 non-null  float64
 18  Gender_Female                      50761 non-null  int64
 19  Gender_Male                        50761 non-null  int64
 20  Type of Travel_Business travel     50761 non-null  int64
 21  Type of Travel_Personal Travel     50761 non-null  int64
 22  Class_Business                     50761 non-null  int64
 23  Class_Eco                          50761 non-null  int64
 24  Class_Eco Plus                     50761 non-null  int64
 25  satisfaction                       50761 non-null  int64
dtypes: float64(1), int64(25)
memory usage: 10.1 MB
```

Finally we observe the data are fully cleaned.

- ❖ Now we check the data dependency.

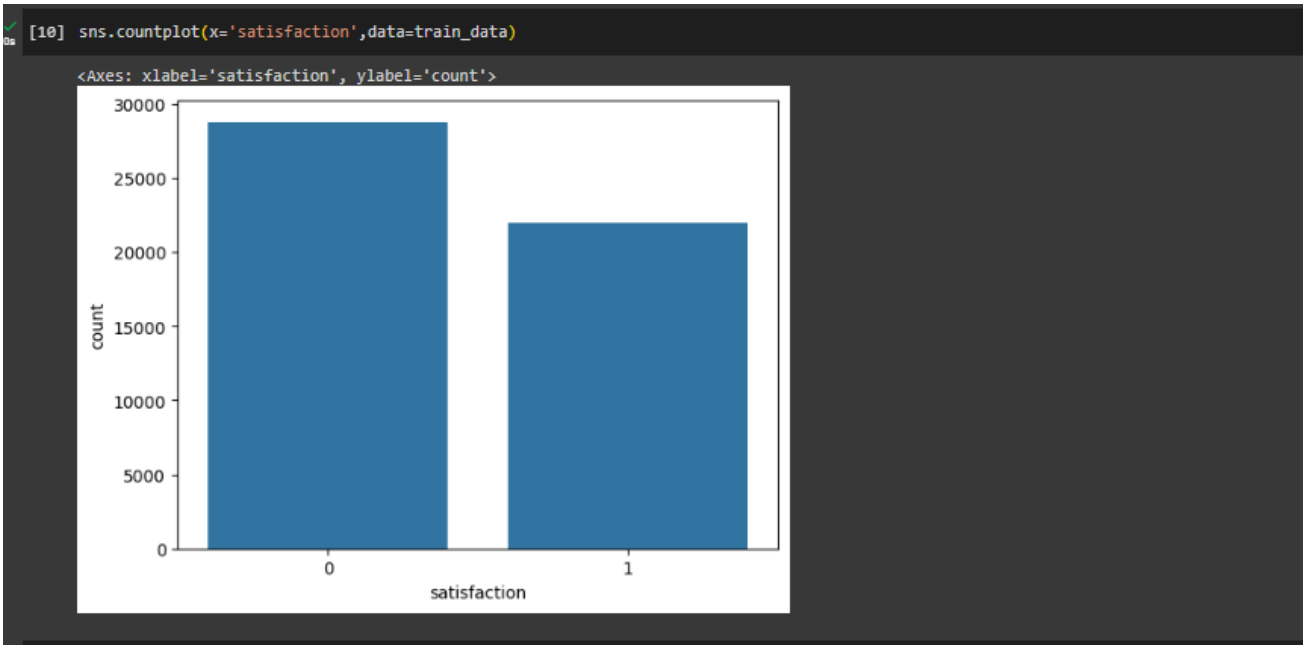
```
plt.figure(figsize=(15,15))
sns.heatmap(train_data[2:8].corr(),annot=True)

<Axes: >
```





We see that data dependent each other.

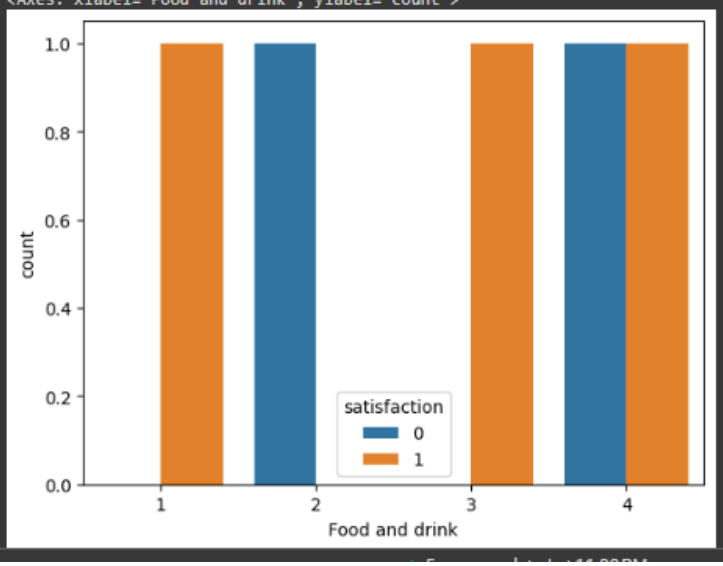
- Visualizing the Airline customer satisfaction like Age, Ease of Online booking, Food and drink, Departure/Arrival time convenient, Check-in service, On-board service etc



As per Visualizing the above graph, customer satisfaction is less than the neutral or dissatisfaction.

```
[11] sns.countplot(hue='satisfaction',x='Food and drink',data=train_data[0:5])
```

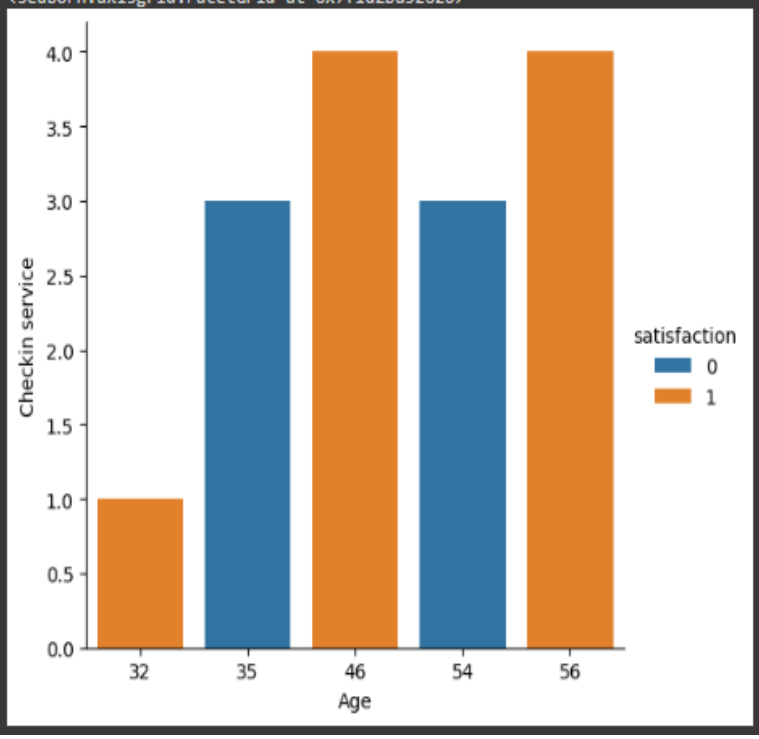
```
<Axes: xlabel='Food and drink', ylabel='count'>
```



As per Visualizing the above graph, customer satisfied with food and drink..

```
sns.catplot(y='Checkin service',hue='satisfaction',x='Age',data=train_data[0:5],kind='bar')
```

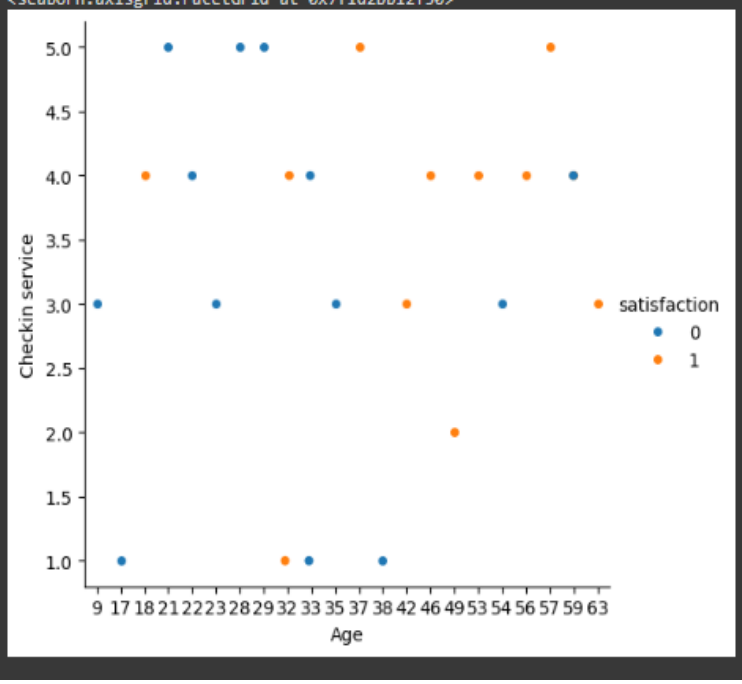
```
<seaborn.axisgrid.FacetGrid at 0x7f1d2ba52620>
```



As per Visualizing the above graph, people satisfied is in age range 32, 46 and 56 in checking service.

```
[13] sns.catplot(y='Checkin service',hue='satisfaction',x='Age',data=train_data[0:25])
```

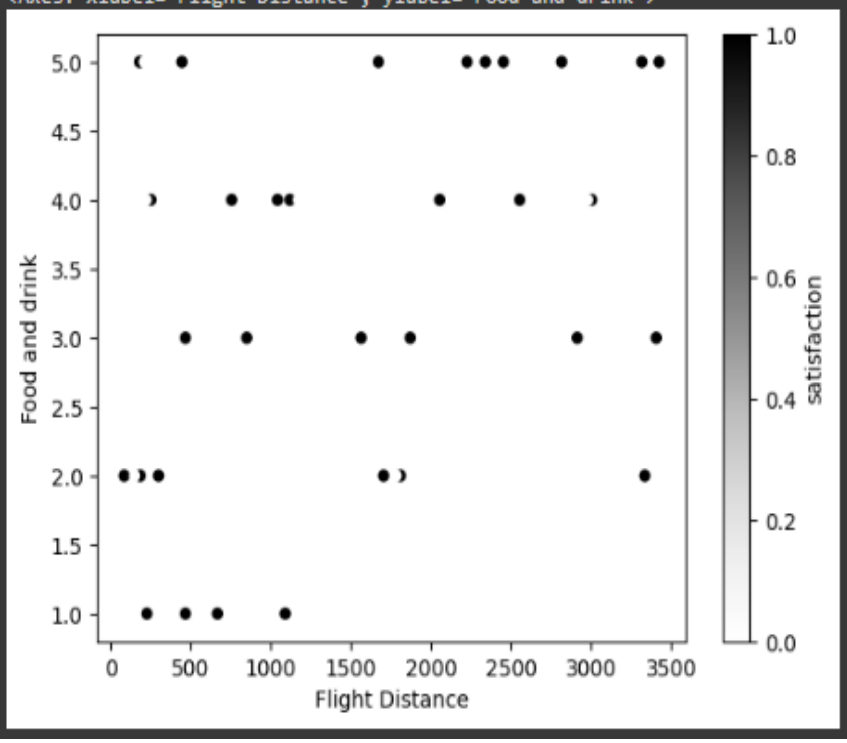
```
<seaborn.axisgrid.FacetGrid at 0x7f1d2bb12f50>
```



We observed that customer have age above 30 is more satisfied in with best checking service rate

```
[14] train_data.iloc[0:69].plot.scatter(x='Flight Distance',y='Food and drink',c='satisfaction')
```

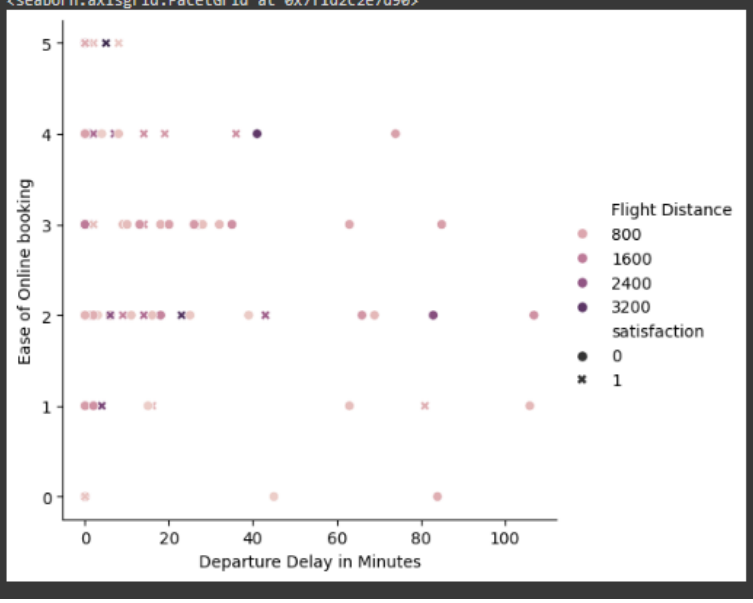
```
<Axes: xlabel='Flight Distance', ylabel='Food and drink'>
```



We observe the above graph, customer satisfied in more distance traveling with good food and drink..

```
[17] sns.relplot(y='Ease of Online booking',hue='Flight Distance',x='Departure Delay in Minutes',style='satisfaction',data=train_data[50:200])
```

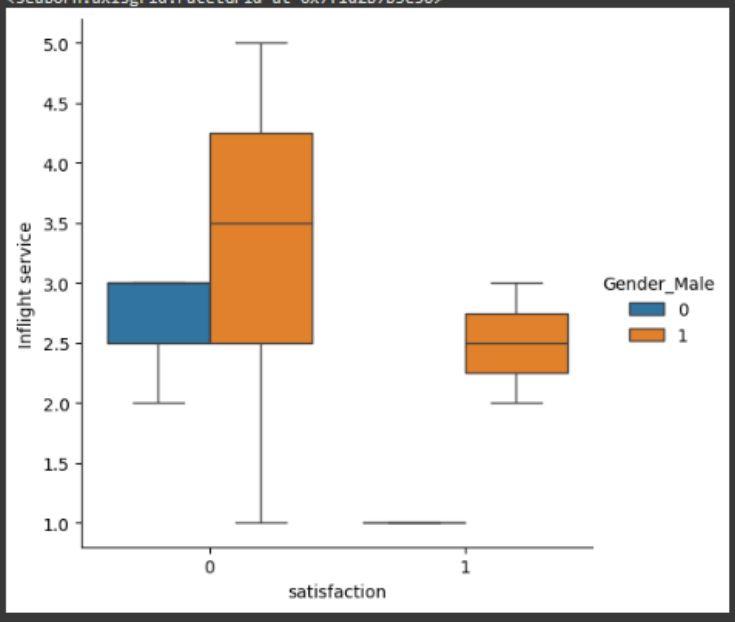
```
<seaborn.axisgrid.FacetGrid at 0x7f1d2c2e7d90>
```



We observe the above graph, customer is satisfy with less departure delay with short distance travel and average ease of online booking service.

```
sns.catplot(hue='Gender_Male',y='Inflight service',x='satisfaction',kind='box',data=train_data[0:10])
```

```
<seaborn.axisgrid.FacetGrid at 0x7f1d2b7b3e50>
```



We observe the above graph, Male customer is more interest than female for traveling in air service.

After visualization of data, we predict Airline customer satisfaction using Machine Learning .

- ❖ Splitting the dataset into dependent(y) & independent(x) sets

```
[23] x=train_data.drop(columns=['satisfaction'])
     y=train_data['satisfaction']
```

- Importing train_test_split from sklearn.model library for splitting the data into train and test sets. (we consider train dataset).

```
[24] from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.85,random_state=0)
```

- Importing logistic regression from sklearn Libaray & then activating the Machine learning Model .Then used regression.fit() to training the model by providing train & test sets as x & y. And then predicted the trained model with help of MLM & the checked score as regression.score(x,y)

```
[25] from sklearn.linear_model import LogisticRegression
     regression=LogisticRegression()

[26] regression.fit(x_train,y_train)

     /usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: 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(
     ▾ LogisticRegression
     LogisticRegression()
```

- Checking the accuracy with help of confusion Matrix.

```
[27] y_predict_regression=regression.predict(x_test)

[28] from sklearn.metrics import confusion_matrix,accuracy_score
     ac=accuracy_score(y_test,y_predict_regression)
     cm=confusion_matrix(y_test,y_predict_regression)

[29] print(ac)
     print(cm)

     0.813263296126067
     [[3470  813]
      [ 609 2723]]
```

In the above model we can see that the accuracy obtained is 81%

- Now applying new algorithm Knn, then checked score.

```
[30] from sklearn.neighbors import KNeighborsClassifier
     knn=KNeighborsClassifier(n_neighbors=5) #where k=5

[31] knn.fit(x_train,y_train)

     ▾ KNeighborsClassifier
     KNeighborsClassifier()

[32] y_predict_knn=knn.predict(x_test)
```

```
[33] ac=accuracy_score(y_test,y_predict_knn)
     cm=confusion_matrix(y_test,y_predict_knn)

[34] print(ac)
     print(cm)

     0.7234405778069599
     [[3392  891]
      [1215 2117]]
```

we can see that the accuracy obtained is 72%

- Now applying new algorithm DecisionTree , then checked score.

```
[35] from sklearn.tree import DecisionTreeClassifier
     tree=DecisionTreeClassifier()

[36] tree.fit(x_train,y_train)

     DecisionTreeClassifier
     DecisionTreeClassifier()

[37] y_predict_tree=tree.predict(x_test)

[38] ac=accuracy_score(y_test,y_predict_tree)
     cm=confusion_matrix(y_test,y_predict_tree)

[39] print(ac)
     print(cm)

     0.9281680892974392
     [[3994  289]
      [ 258 3074]]
```

we can see that the accuracy obtained is 92%,is best for test data set.

- Now applying new algorithm RandomForest , then checked score.

```
[40] from sklearn.ensemble import RandomForestClassifier
     random=DecisionTreeClassifier()

[41] random.fit(x_train,y_train)
     y_predict_random=random.predict(x_test)

[42] ac=accuracy_score(y_test,y_predict_random)
     cm=confusion_matrix(y_test,y_predict_random)
     print(ac)
     print(cm)

     0.9273801707156927
     [[3995  288]
      [ 265 3067]]
```

we can see that the accuracy obtained with Random forest and decision tree both are same approx.92%

- We want to check with other algorithms(AdaBoost) for best accuracy

```
[ ] from sklearn.ensemble import AdaBoostClassifier
    ada_classifier = AdaBoostClassifier(DecisionTreeClassifier(max_depth=1),n_estimators=20)

[ ] ada_classifier.fit(x_train,y_train)
    y_predict_ada=ada_classifier.predict(x_test)

[ ] ac=accuracy_score(y_test,y_predict_ada)
    cm=confusion_matrix(y_test,y_predict_ada)
    print(ac)
    print(cm)

    0.9054497701904136
    [[3981  302]
     [ 418 2914]]
```

We see the accuracy is good but less than Decision Tree and Random forest algorithms.

- Now we compare all algorithms with accuracy

Algorithms	accuracy
Logistic regression	81%
KNN	72%
Ada Boost	90%
Random Forest classifier	92.7%
Decision Tree classifier	92.8%

Random Forest and Decision Tree Classifier machine algorithms is better than KNN and Logistic regression.

- Now recalling the test data set.
- Loading the csv-dataset in the variable name ‘test_data’ Then viewing the data with test_data.head()

```
[ ] test_data.head()
```

	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding	Seat comfort	Inflight entertainment	...	Cleanliness	Departure Delay in Minutes	Arrival Delay in Minutes	Gender_Female	Gender_Male	Travel_B
0	46	1622	1	1	1	1	5	5	5	4	...	5	276	270.0	0	1	
1	45	552	3	1	3	4	4	5	5	5	...	5	0	0.0	1	0	
2	52	435	2	2	2	2	3	4	5	4	...	4	0	0.0	0	1	
3	41	655	2	5	2	3	4	2	1	4	...	4	0	0.0	1	0	
4	39	337	2	0	1	3	5	1	5	5	...	5	0	0.0	1	0	

5 rows × 25 columns

- Splitting into test & train sets as x1_test & x1_train. Then we find the Airline customer satisfaction using Machine Learning(Decision Tree classifier)

```
[ ] from sklearn.tree import DecisionTreeClassifier
    tree_test=DecisionTreeClassifier()

[ ] tree_test.fit(x_train,y_train)

    DecisionTreeClassifier
    DecisionTreeClassifier()
```

Applying Decision Tree classifier algorithms for predictions.

```
y_predict_test

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

[ ] print('Airline customer satisfaction prediction ',y_predict_test)

    Airline customer satisfaction prediction  [1 1 0 ... 0 0 0]
```

Conclusion:- In this test data set we analysed the data we found the maximum customer are neutral or dissatisfied with airline service.

Thank you