2448513_Deshmukh_Pratik_Bhushanrao_ML_Lab-2

Name: Deshmukh Pratik Bhushanrao
Roll no.: 2448513
Subject: Machine Learning
Subject code: MAI171

Inferance: Importing all the essential libraries for our task

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

Inferance: Importing the dataset to into the variable of dataframe

In [ ]:
```
df = pd.read_csv(r'Invistico_Airline - Invistico_Airline.csv')
df.head()
```

Out[ ]:

	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departu time c
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	

5 rows × 23 columns

Inferance: In the next line we Find the basic description of the dataset

In [ ]:
```
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
```

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

Out[ ]:

	Age	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink
count	129880.000000	129880.000000	129880.000000	129880.000000	129880.000000
mean	39.427957	1981.409055	2.838597	2.990645	2.851994
std	15.119360	1027.115606	1.392983	1.527224	1.443729
min	7.000000	50.000000	0.000000	0.000000	0.000000
25%	27.000000	1359.000000	2.000000	2.000000	2.000000
50%	40.000000	1925.000000	3.000000	3.000000	3.000000
75%	51.000000	2544.000000	4.000000	4.000000	4.000000
max	85.000000	6951.000000	5.000000	5.000000	5.000000

Inferance: Checking the null values in the dataset

In [ ]:
```
df.isna().sum()
```

Out[ ]:
```
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
```

Inferance: Filling the null values the the "Arrival Delay in Minutes" by taking the mean of the "Arrival Delay in Minutes" and replacing it with the missing values in the dataset

In [ ]:
```
df['Arrival Delay in Minutes'].fillna(df['Arrival Delay in Minutes'].mean(), inp
```

```
C:\Users\prati\AppData\Local\Temp\ipykernel_28500\613989052.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.meth
od({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to pe
rform the operation inplace on the original object.


  df['Arrival Delay in Minutes'].fillna(df['Arrival Delay in Minutes'].mean(), in
place=True)
```

Inferance: Again checking the null values in the dataset

In [ ]:
```
df.isna().sum()
```

Out[ ]:
```
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
```

Feature Engineering

Creating Rating column

Inferance: The following function will return value between 0-2.5 if its not-satisfied and 2.5-5 if satisfied

In [ ]:
```
from random import uniform
def random_rating(satisfaction):
    rating_stack = []
    for rating in satisfaction:
        if rating == 'satisfied':
            rating_stack.append(round(uniform(2.5, 5), 1))
        else:
            rating_stack.append(round(uniform(0, 2.5), 1))

    return rating_stack
```

Inferance: Creating new column called Rating which uses above function creating new column

In [ ]:
```
rating = random_rating(df['satisfaction'])
df["Rating"] = rating
df.head()
```

Out[ ]:

	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departu time c
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	

5 rows × 24 columns

Creating Repeat Purchase

Inferance: The following function will return 1 if customer rating is above 3 else it'll return 0

In [ ]:
```
def random_repeat_purchase(ratings):
    repeat_purchase_stack = []
    for rating in ratings:
        if rating >= 3:
            repeat_purchase_stack.append(1)
        else:
            repeat_purchase_stack.append(0)


    return repeat_purchase_stack
```

Inferance: The following lines will create new column called 'Repeat Purchaser' by using above function

In [ ]:
```
repeat_purchaser = random_repeat_purchase(df['Rating'])
df["Repeat Purchaser"] = repeat_purchaser
df.head()
```

Out[ ]:

	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departu time c
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	

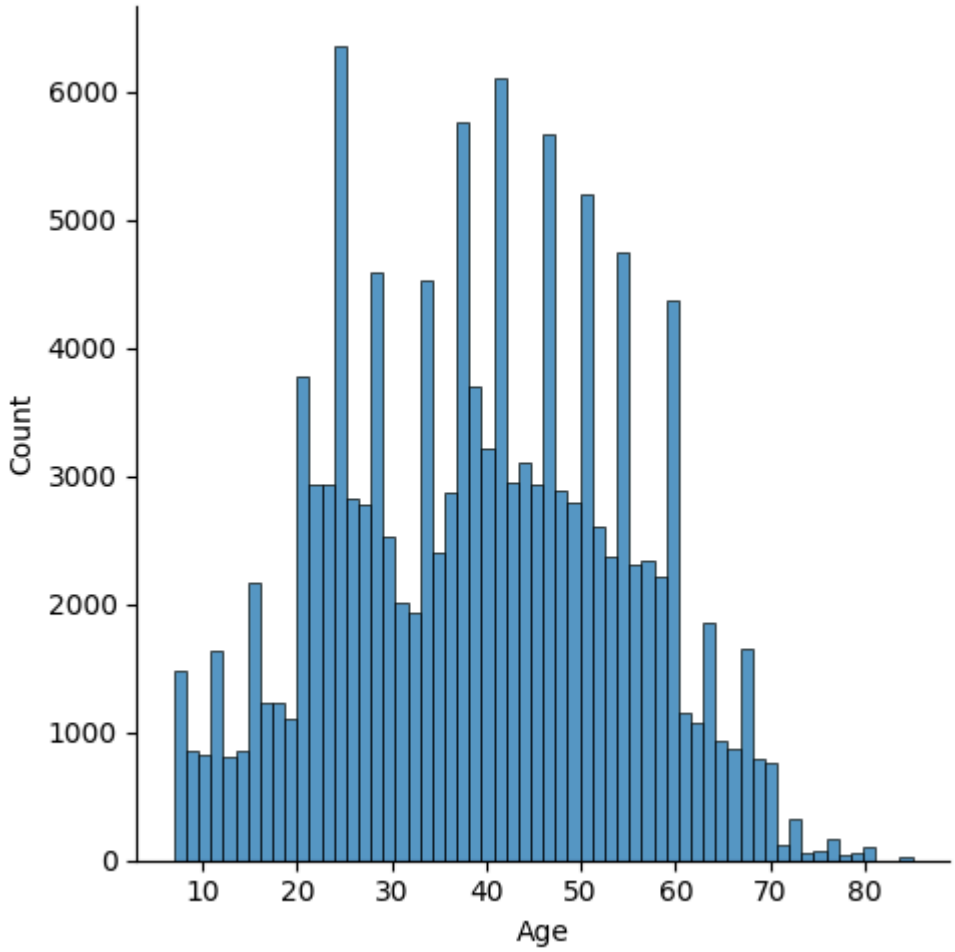
5 rows × 25 columns

Tasks

Inferance: showing the overall distribution of the ages in our dataset

In [ ]:
```
sns.displot(df.Age, bins=60)
```

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x146b0b59550>
```



Creating Bins

Inferance: The following line will create different bins starting from "7-17", "18-35", "36-57", "58+" based upon the age of customer

In [ ]:
```
bin_enges = [7, 18, 36, 58, np.inf]

df['Age_bin'] = pd.cut(df['Age'], bins=bin_enges, labels=["7-17", "18-35", "36-5
```

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

Out[ ]:

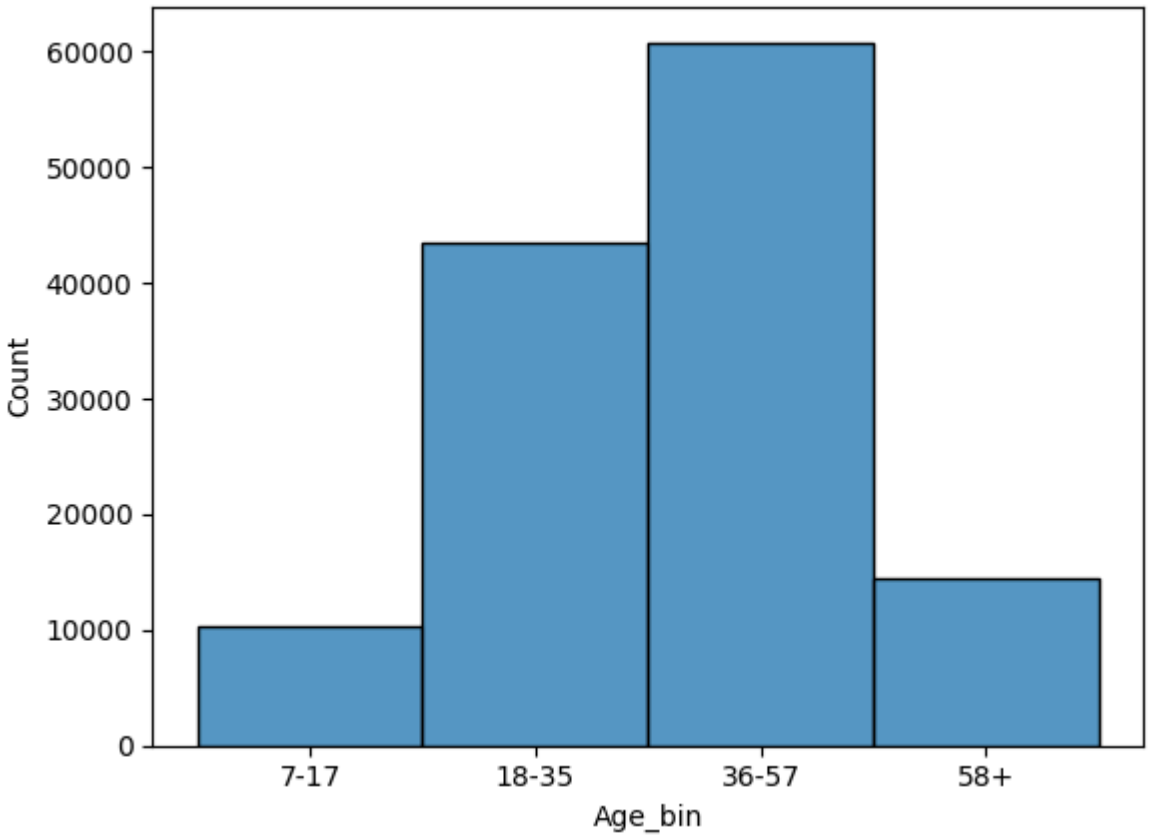
	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departu time c
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	

5 rows × 26 columns

Inferance: Plotting histogram for showing distribution

In [ ]:
```
sns.histplot(df.Age_bin)
```

Out[ ]: <Axes: xlabel='Age_bin', ylabel='Count'>



Handling categorical variables gender and Customer Type

Inferance: In the following lines we'll handle categorical variables 'gender' and 'customer type'

In [ ]:
```
df = pd.concat([df, pd.get_dummies(df[['Customer Type', 'Gender']], drop_first=T
df.head()
```

Out[ ]:

	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departu time c
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	

5 rows × 28 columns

Finding relationship between variables

Inferance: In the next line we'll find relationship between Age of the customer and the Ratings given by the customer.

In [ ]:
```
df['satisfaction_dum'] = pd.get_dummies(df.satisfaction, drop_first=True, dtype=
df[['Age', 'Rating']].cov()
```

Out[ ]:

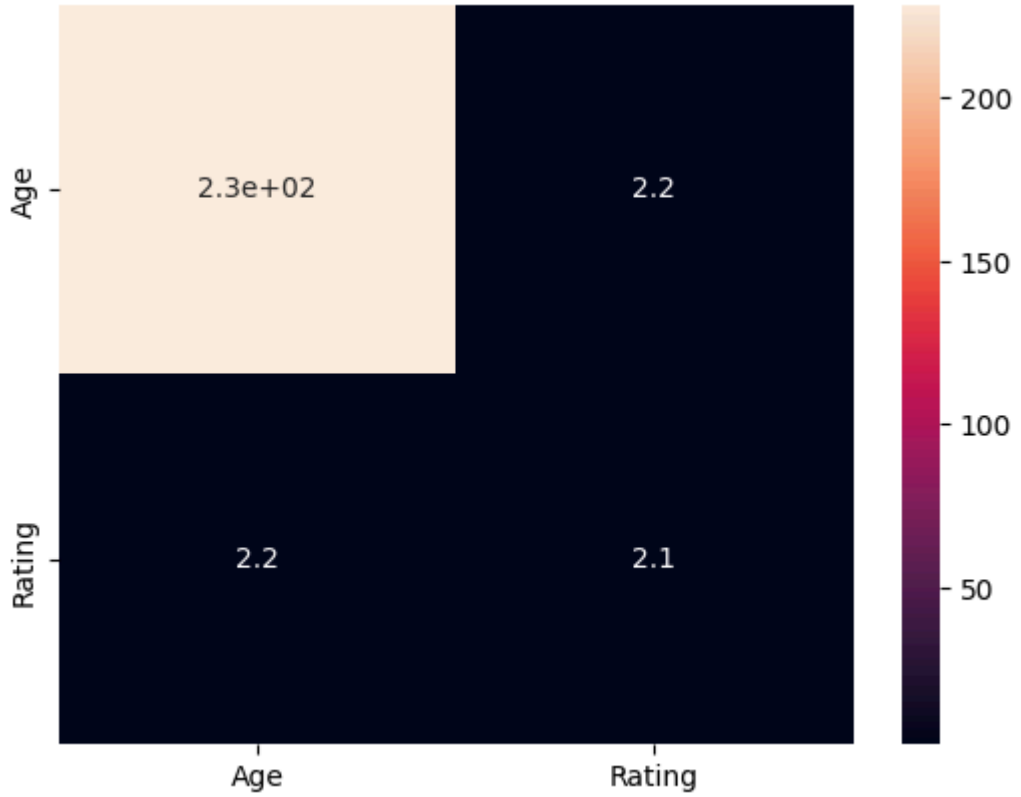
	Age	Rating
Age	228.595045	2.180268
Rating	2.180268	2.072101

Plotting covariance matrix

Inferance: In the following line we'll create covariance matrix for finding covariance between Age of the customer and the Ratings given by the customer

In [ ]:
```
sns.heatmap(df[['Age', 'Rating']].cov(), annot=True)
```

Out[ ]: <Axes: >



Association between Gender and Repeat Purchase

Inferance: For finding relationship between 2 categorical variables we use "Chi-Square Test of Independence"

In [ ]:
```
from sklearn.feature_selection import chi2
import math
chi_2 = chi2(df[['Gender_Male' ]], df['Repeat Purchaser'])
math.sqrt(chi_2[0])
```

```
C:\Users\prati\AppData\Local\Temp\ipykernel_28500\3030085808.py:4: DeprecationWar
ning: Conversion of an array with ndim > 0 to a scalar is deprecated, and will er
ror in future. Ensure you extract a single element from your array before perform
ing this operation. (Deprecated NumPy 1.25.)
  math.sqrt(chi_2[0])
```

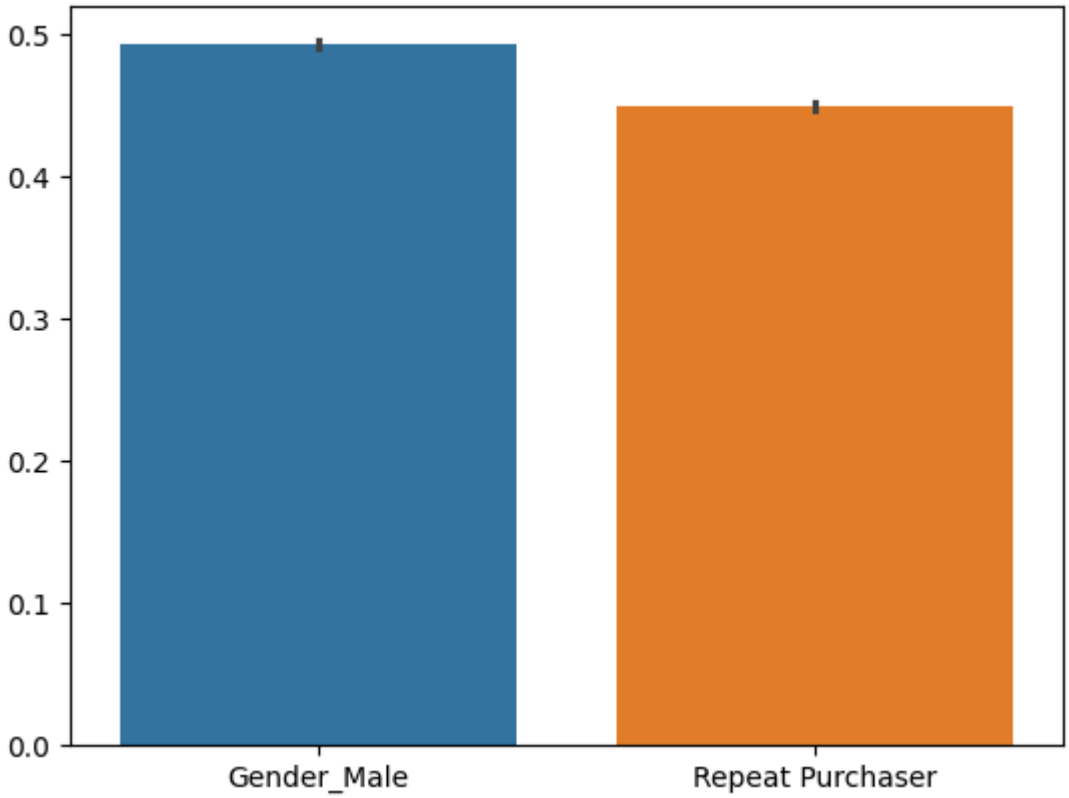
Out[ ]: 44.68072461596668

Inferance: Plotting a barplot for finding association between Gender and the repeat purchase

Note: the bar plot is not a suitable for this task

In [ ]:
```
sns.barplot(df[['Gender_Male', 'Repeat Purchaser']])
```

Out[ ]: <Axes: >



Comparing ratings of repeat purchasers and non repeat purchasers

Inferance: for this we'll perform independant t-test

In [ ]:
```
repeat_purchasor, non_repeat_purchasor = df[df['Repeat Purchaser'] == 1]['Rating


from scipy.stats import ttest_ind
ttest_ind(repeat_purchasor, non_repeat_purchasor, equal_var=True)
```

Out[ ]:
```
TtestResult(statistic=584.6607028273936, pvalue=0.0, df=129878.0)
```

Inferance: we'll state our hyplthesis, calculate the test statistic and p-value, and interpret the results

In [ ]:
```
Null_Hypothesis = "Repeat Purchaser has no effect on Rating"
Alternative_Hypothesis = "Repeat Purchaser has an effect on Rating"
```

In [ ]:
```

```