Name: Pranav Kumar

PRN: 1032200232

Exp. No.: 4

Class: Final Year B.Tech

Aim: To Perform Exploratory Data Analysis on the given dataset.

Explanation:

Exploratory Data Analysis (EDA) is a crucial initial step in data analysis, aimed at gaining insights and understanding the dataset before diving into modeling or more advanced analyses. Here are five key points to explain the importance and goals of EDA:

1. Data Understanding: EDA helps you understand the dataset's structure, including its dimensions, variables, and data types. It reveals potential challenges, such as missing data, outliers, or inconsistencies, which need to be addressed during preprocessing.

2. Pattern Discovery: EDA enables the identification of patterns, trends, and relationships within the data. Visualization tools and statistical summaries help reveal data distributions, correlations, and potential clusters or groups.

3. Insightful Visualizations: EDA often involves creating visualizations (e.g., histograms, scatter plots, box plots) that provide intuitive insights into the data. These visualizations help communicate findings and patterns to stakeholders and guide further analysis.

Operations Preformed are –

1. Handling Missing data
2. Filtering Data
3. Grouping Data
4. Finding the outliers
5. Etc.

Part A:

Program:

```
import pandas as pd

#Taking data
data = {'Name': ["Sam", "Kia", "Jack", "lilly", "Riya", "Keshav", "Rose"],
      'Age': [12, 13, 14, 13, 12, 14, 13],
      'Gender': ['M', 'F', 'M', 'F', 'F', 'M', 'F'],
      'Marks': [98, 97, 'Nan', 65, 74, 'Nan', 66]}
print(data)

#Making Dataframe
df = pd.DataFrame(data)
print(df)

#Checking null values
#print(df.isnull().sum())
#print(df.info())
#print(df.describe())

#calulate Average
c = avg = 0
for ele in df['Marks']:
   if str(ele).isnumeric():
      c += 1
      avg += ele
avg /= c
print('avg',avg)
print('c',c)

#Replace the null values with the average values
df = df.replace(to_replace = "Nan",
            value = avg)
print(df)

#objects
print(df.info())

#convert object to string
df['Gender'] = df['Gender'].map({'M' : "Male",
                        'F' : "Female"}).astype("string")    #astype is type conversion from object
to string
print(df)
print(df.info())
```

```python
#Data filtering
df = df[df['Marks'] >= 75]
print(df)

#data filtering - remove column
df = df.drop(['Age'], axis=1)
print(df)

#Add ID column to the existing table
data['ID'] = [101, 103, 105, 104, 102, 107, 106]
data1 = pd.DataFrame(data) #create dataframe again
print(data1)

#create table 2 with one column similar
data2 = pd.DataFrame({'ID': [101, 103, 105, 104, 102, 107, 106],
                'Fee Status': ["Paid", "Unpaid", "Unpaid", "Unpaid", "Paid", "Unpaid", "Paid"]})
print(data2)

#merge the two tables using ID as the merging column
data3 = pd.merge(data1, data2, on="ID")
print(data3)

#make group of the data whose age is 13
grouped = data3.groupby('Age')
print(grouped.get_group(13))
```

Output:

```
{'Name': ['Sam', 'Ria', 'Jack', 'Lilly', 'Riya', 'Keshav', 'Rose'], 'Age': [12, 13, 14, 13, 12, 14, 13], 'Gender': ['M', 'F', 'M', 'F', 'F', 'M', 'F'], 'Marks': [98, 97, 'Nan'
     Name  Age Gender Marks
0     Sam   12      M    98
1     Ria   13      F    97
2    Jack   14      M   Nan
3   Lilly   13      F    65
4    Riya   12      F    74
5  Keshav   14      M   Nan
6    Rose   13      F    66
avg 80.0
c 5
     Name  Age Gender  Marks
0     Sam   12      M   98.0
1     Ria   13      F   97.0
2    Jack   14      M   80.0
3   Lilly   13      F   65.0
4    Riya   12      F   74.0
5  Keshav   14      M   88.0
6    Rose   13      F   66.0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    7 non-null      object
 1   Age     7 non-null      int64
 2   Gender  7 non-null      object
 3   Marks   7 non-null      float64
```

```
RangeIndex: 7 entries, 0 to 6
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    7 non-null      object
 1   Age     7 non-null      int64
 2   Gender  7 non-null      object
 3   Marks   7 non-null      float64
dtypes: float64(1), int64(1), object(2)
memory usage: 356.0+ bytes
None
     Name  Age  Gender  Marks
0     Sam   12    Male   98.0
1     Kia   13  Female   97.0
2    Jack   14    Male   80.0
3   lilly   13  Female   65.0
4    Riya   12  Female   74.0
5  Keshav   14    Male   80.0
6    Rose   13  Female   66.0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    7 non-null      object
 1   Age     7 non-null      int64
 2   Gender  7 non-null      string
 3   Marks   7 non-null      float64
dtypes: float64(1), int64(1), object(1), string(1)
```

```
     Name  Age Gender Marks   ID
0     Sam   12      M    98  101
1     Kia   13      F    97  103
2    Jack   14      M   Nan  105
3   lilly   13      F    65  104
4    Riya   12      F    74  102
5  Keshav   14      M   Nan  107
6    Rose   13      F    66  106
    ID Fee Status
0  101       Paid
1  103     Unpaid
2  105     Unpaid
3  104     Unpaid
4  102       Paid
5  107     Unpaid
6  106       Paid
     Name  Age Gender Marks   ID Fee Status
0     Sam   12      M    98  101       Paid
1     Kia   13      F    97  103     Unpaid
2    Jack   14      M   Nan  105     Unpaid
3   lilly   13      F    65  104     Unpaid
4    Riya   12      F    74  102       Paid
5  Keshav   14      M   Nan  107     Unpaid
6    Rose   13      F    66  106       Paid
    Name  Age Gender Marks   ID Fee Status
1    Kia   13      F    97  103     Unpaid
3  lilly   13      F    65  104     Unpaid
6   Rose   13      F    66  106       Paid
```

Part B:

Program:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np



#open CSV file

df = pd.read_csv("titanic.csv")

print(df)



print(df.head())



df2 = df[['Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']]

print(df2.head())



print(df2.isnull().sum())
```

```python
print(df2.info())



updated_df1 = df2.dropna(axis=1)

print(updated_df1.info())



updated_df2 = df2.dropna(axis=0)

print(updated_df2.info())



print('skew', df2['Age'].skew())



#updated_df3 = df2

#updated_df3['Age'] = updated_df3['Age'].fillna(updated_df3['Age'].mean())

#print(updated_df3.info())



sample = [15,101,18,7,13,16,11,21,5,15,10,9,-1]

print('\n\nDifferent prog:')
```

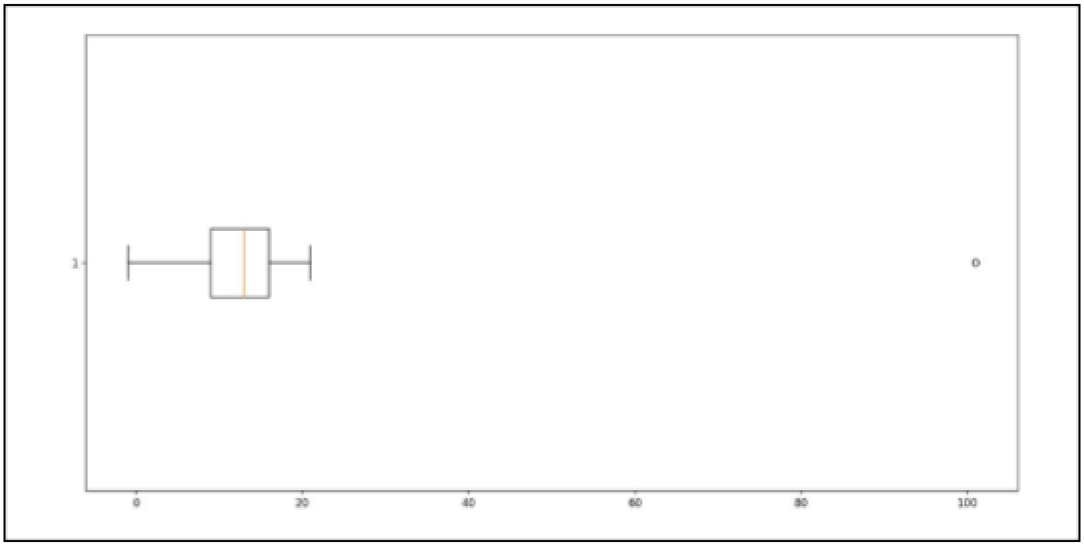
```python
print('mean',np.mean(sample))

print('median',np.median(sample))

print("sample",sample)

print("Q2 quantile of sample", np.median(sample))

print("Q1 quantile of sample", np.quantile(sample, .25))

print("Q3 quantile of sample", np.quantile(sample, .75))

plt.boxplot(sample, vert=False)

plt.show()
```

Output:

```
     PassengerId  Survived  Pclass  ...     Fare Cabin  Embarked
0              1         0       3  ...   7.2500   NaN         S
1              2         1       1  ...  71.2833   C85         C
2              3         1       3  ...   7.9250   NaN         S
3              4         1       1  ...  53.1000  C123         S
4              5         0       3  ...   8.0500   NaN         S
..           ...       ...     ...  ...      ...   ...       ...
886          887         0       2  ...  13.0000   NaN         S
887          888         1       1  ...  30.0000   B42         S
888          889         0       3  ...  23.4500   NaN         S
889          890         1       1  ...  30.0000  C148         C
890          891         0       3  ...   7.7500   NaN         Q

[891 rows x 12 columns]
   PassengerId  Survived  Pclass  ...     Fare Cabin  Embarked
0            1         0       3  ...   7.2500   NaN         S
1            2         1       1  ...  71.2833   C85         C
2            3         1       3  ...   7.9250   NaN         S
3            4         1       1  ...  53.1000  C123         S
4            5         0       3  ...   8.0500   NaN         S

[5 rows x 12 columns]
   Survived  Pclass     Sex   Age  SibSp  Parch     Fare
0         0       3    male  22.0      1      0   7.2500
1         1       1  female  38.0      1      0  71.2833
2         1       3  female  26.0      0      0   7.9250
3         1       1  female  35.0      1      0  53.1000
4         0       3    male  35.0      0      0   8.0500
```

```
Survived      0
Pclass        0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Sex       891 non-null    object
 3   Age       714 non-null    float64
 4   SibSp     891 non-null    int64
 5   Parch     891 non-null    int64
 6   Fare      891 non-null    float64
dtypes: float64(2), int64(4), object(1)
memory usage: 48.9+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
```

```
dtypes: float64(1), int64(4), object(1)
memory usage: 41.9+ KB
None
<class 'pandas.core.frame.DataFrame'>
Index: 714 entries, 0 to 890
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  714 non-null    int64
 1   Pclass    714 non-null    int64
 2   Sex       714 non-null    object
 3   Age       714 non-null    float64
 4   SibSp     714 non-null    int64
 5   Parch     714 non-null    int64
 6   Fare      714 non-null    float64
dtypes: float64(2), int64(4), object(1)
memory usage: 44.6+ KB
None
skew 0.38910778230082704


Different prog:
mean 18.46153846153846
median 13.0
sample [15, 101, 10, 7, 13, 16, 11, 21, 5, 15, 10, 9, -1]
Q2 quantile of sample 13.0
Q1 quantile of sample 9.0
Q3 quantile of sample 16.0
```



Conclusion:

From this experiment we learnt about Machine Leaning and the different data processing techniques to make the data usage more efficient. We collected data and performed different data sorting and data analysis techniques. Code for the same was done successfully.