Date:

EXPERIMENT-4

AIM:

Load any public dataset and understand the basic information about data and statistical summary of the dataset.

REQUIREMENTS:

1. Dataset:

UCI Heart Disease Data

This is a multivariate type of dataset. It is composed of 14 attributes which are age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina, oldpeak — ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels and Thalassemia.

2. Libraries for exploration:

Pandas

Pandas is a python library used for data manipulation and statistical analysis. It is a fast and easy to use open-source library that enables several data manipulation tasks. These include merging, reshaping, wrangling, statistical analysis and much more. In this post, we will discuss how to calculate summary statistics using the Pandas library.

PROCEDURE:

STEP 1: Import required libraries using import command in Google colab .

STEP 2: Load the data into google colab using read_csv command.

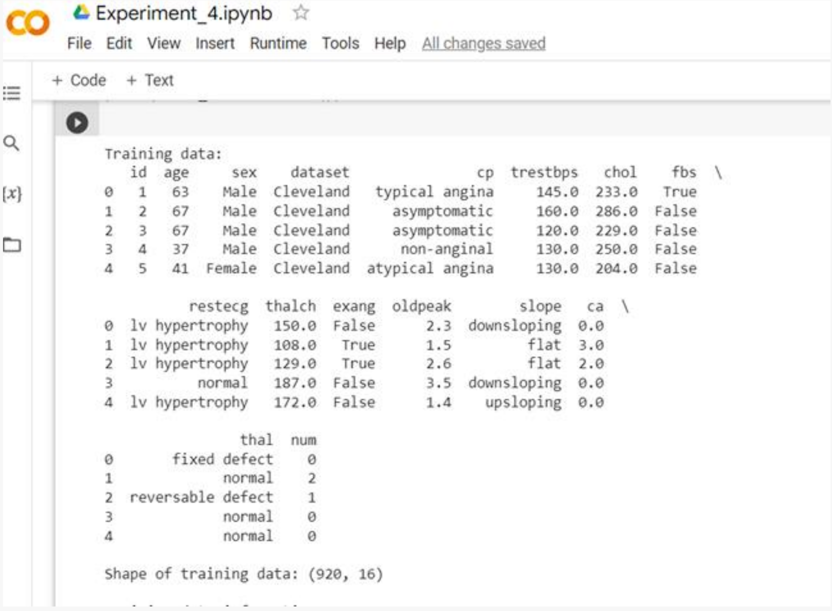
STEP 3: Exploring the data scatter and do statistical summary.

CODE:

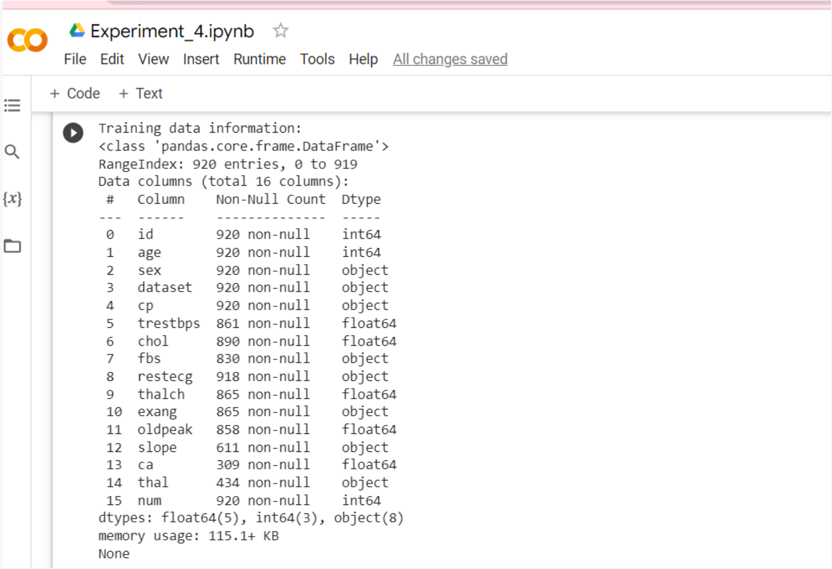
```
import pandas as pd
```

```
# Load the Heart_disease dataset
train_data = pd.read_csv('heart_disease_uci.csv')

# Display the first few rows of the training data
print("Training data:")
print(train_data.head())

# Get the shape of the training data (number of rows, number of columns)
print("\nShape of training data:", train_data.shape)

# Get information about the columns and data types in the training data
print("\nTraining data information:")
print(train_data.info())

# Get summary statistics of the numeric columns in the training data
print("\nSummary statistics of training data:")
print(train_data.describe())
```

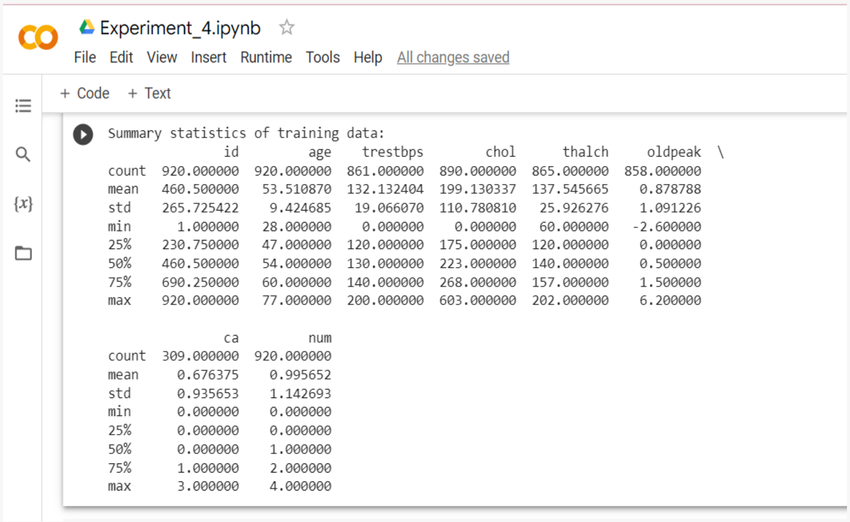
OUTPUT:



```
Training data:
   id  age     sex   dataset               cp  trestbps   chol    fbs  \
0   1   63    Male  Cleveland   typical angina     145.0  233.0   True
1   2   67    Male  Cleveland     asymptomatic     160.0  286.0  False
2   3   67    Male  Cleveland     asymptomatic     120.0  229.0  False
3   4   37    Male  Cleveland      non-anginal     130.0  250.0  False
4   5   41  Female  Cleveland  atypical angina     130.0  204.0  False

          restecg  thalch  exang  oldpeak        slope   ca  \
0  lv hypertrophy   150.0  False      2.3  downsloping  0.0
1  lv hypertrophy   108.0   True      1.5         flat  3.0
2  lv hypertrophy   129.0   True      2.6         flat  2.0
3          normal   187.0  False      3.5  downsloping  0.0
4  lv hypertrophy   172.0  False      1.4    upsloping  0.0

                thal  num
0       fixed defect    0
1             normal    2
2  reversable defect    1
3             normal    0
4             normal    0

Shape of training data: (920, 16)
```



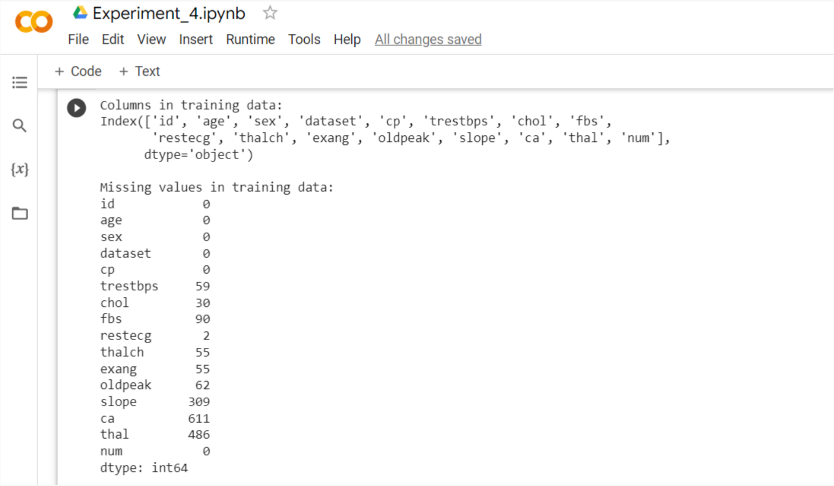
```
Training data information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 920 entries, 0 to 919
Data columns (total 16 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   id        920 non-null    int64  
 1   age       920 non-null    int64  
 2   sex       920 non-null    object 
 3   dataset   920 non-null    object 
 4   cp        920 non-null    object 
 5   trestbps  861 non-null    float64
 6   chol      890 non-null    float64
 7   fbs       830 non-null    object 
 8   restecg   918 non-null    object 
 9   thalch    865 non-null    float64
 10  exang     865 non-null    object 
 11  oldpeak   858 non-null    float64
 12  slope     611 non-null    object 
 13  ca        309 non-null    float64
 14  thal      434 non-null    object 
 15  num       920 non-null    int64  
dtypes: float64(5), int64(3), object(8)
memory usage: 115.1+ KB
None
```



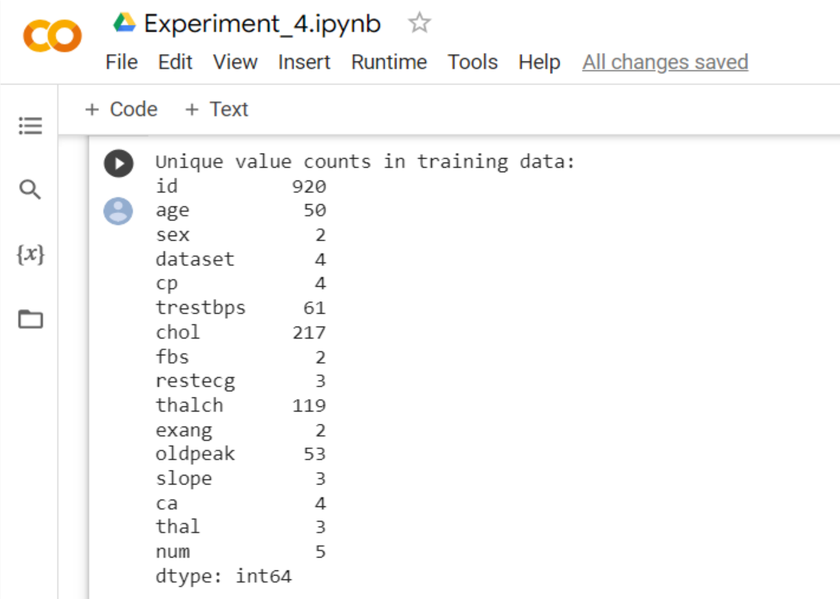
```
Summary statistics of training data:
               id         age    trestbps        chol      thalch     oldpeak  \
count  920.000000  920.000000  861.000000  890.000000  865.000000  858.000000   
mean   460.500000   53.510870  132.132404  199.130337  137.545665    0.878788   
std    265.725422    9.424685   19.066070  110.780810   25.926276    1.091226   
min      1.000000   28.000000    0.000000    0.000000   60.000000   -2.600000   
25%    230.750000   47.000000  120.000000  175.000000  120.000000    0.000000   
50%    460.500000   54.000000  130.000000  223.000000  140.000000    0.500000   
75%    690.250000   60.000000  140.000000  268.000000  157.000000    1.500000   
max    920.000000   77.000000  200.000000  603.000000  202.000000    6.200000   

               ca         num  
count  309.000000  920.000000  
mean     0.676375    0.995652  
std      0.935653    1.142693  
min      0.000000    0.000000  
25%      0.000000    0.000000  
50%      0.000000    1.000000  
75%      1.000000    2.000000  
max      3.000000    4.000000  
```

```
# Display the column names in the training data
print("Columns in training data:")
print(train_data.columns)

# Check for missing values in the training data
print("\nMissing values in training data:")
print(train_data.isnull().sum())


# Count the number of unique values in each column of the training data
print("\nUnique value counts in training data:")
print(train_data.nunique())


# Check the data types of each column in the training data
print("\nData types in training data:")
print(train_data.dtypes)

# Check the distribution of other categorical variables
print("\nSex distribution:")
print(train_data['sex'].value_counts())

print("\nChest pain type distribution:")
print(train_data['cp'].value_counts())

print("\nFasting blood sugar distribution:")
print(train_data['fbs'].value_counts())

print("\nResting electrocardiographic results distribution:")
print(train_data['restecg'].value_counts())

print("\nExercise induced angina distribution:")
print(train_data['exang'].value_counts())

print("\nNumber of major vessels distribution:")
print(train_data['ca'].value_counts())

print("\nThalassemia distribution:")
print(train_data['thal'].value_counts())
```

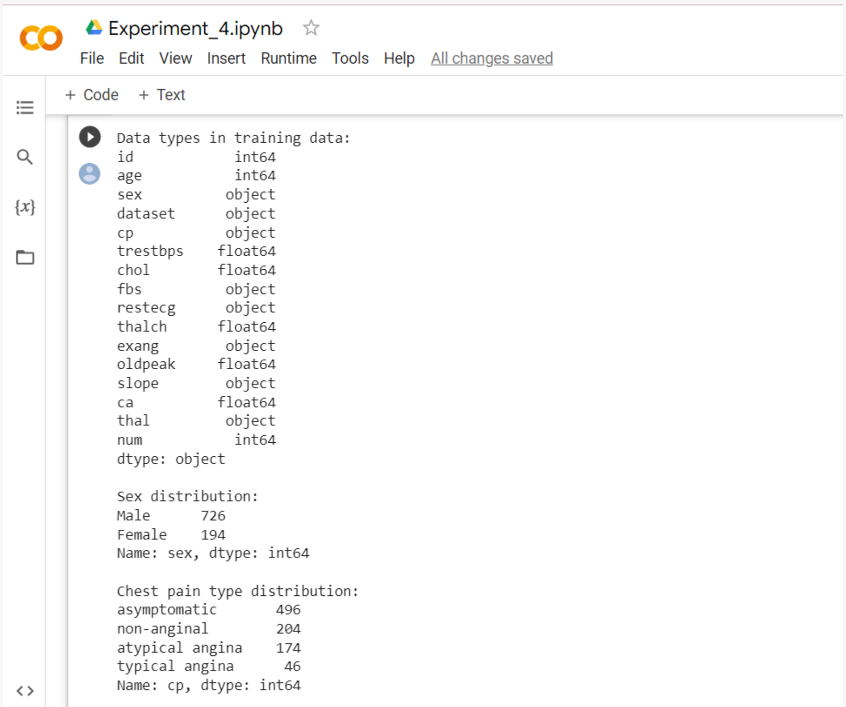
OUTPUT:



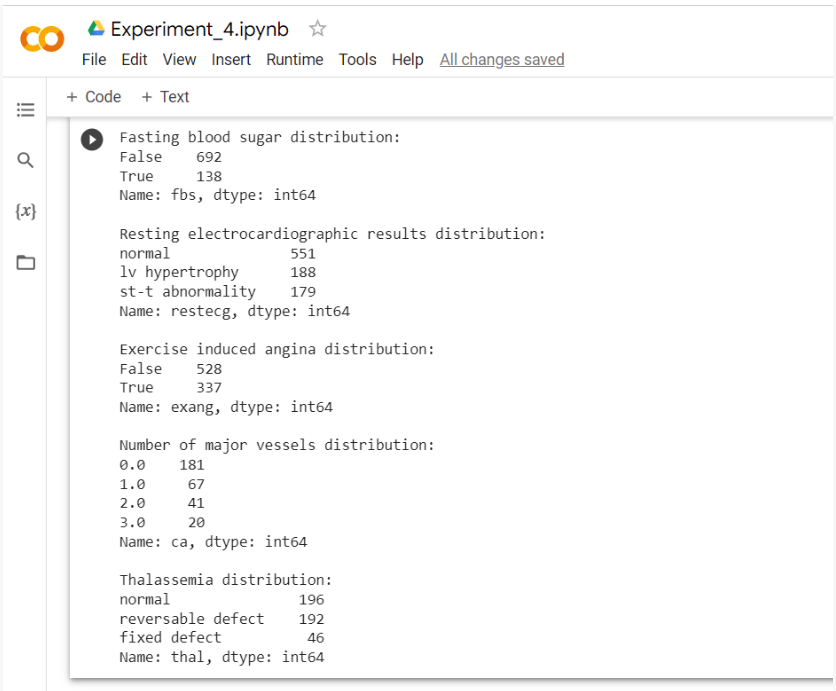
```
Columns in training data:
Index(['id', 'age', 'sex', 'dataset', 'cp', 'trestbps', 'chol', 'fbs',
       'restecg', 'thalch', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num'],
      dtype='object')

Missing values in training data:
id            0
age           0
sex           0
dataset       0
cp            0
trestbps     59
chol         30
fbs          90
restecg       2
thalch       55
exang        55
oldpeak      62
slope       309
ca          611
thal        486
num           0
dtype: int64
```



```
Unique value counts in training data:
id          920
age          50
sex           2
dataset       4
cp            4
trestbps     61
chol        217
fbs           2
restecg       3
thalch      119
exang         2
oldpeak      53
slope         3
ca            4
thal          3
num           5
dtype: int64
```



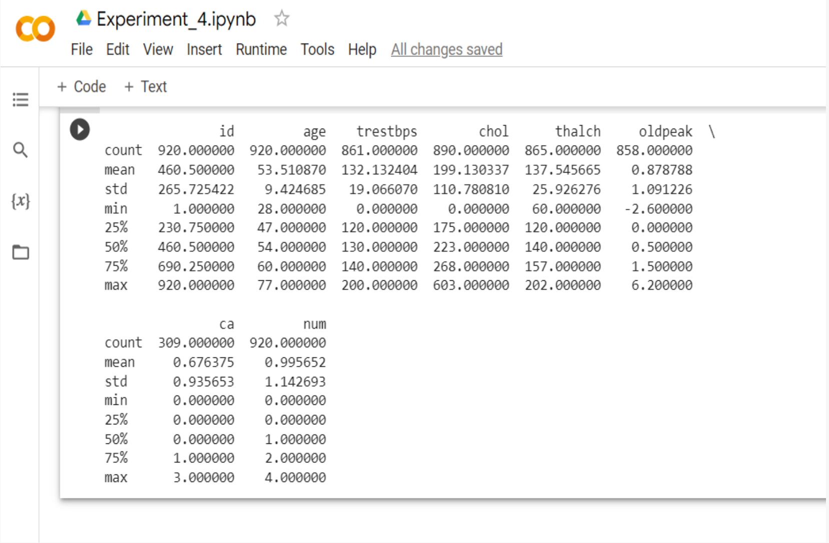
```
Data types in training data:
id            int64
age           int64
sex          object
dataset      object
cp           object
trestbps    float64
chol        float64
fbs          object
restecg      object
thalch      float64
exang        object
oldpeak     float64
slope        object
ca          float64
thal         object
num           int64
dtype: object

Sex distribution:
Male      726
Female    194
Name: sex, dtype: int64

Chest pain type distribution:
asymptomatic       496
non-anginal        204
atypical angina    174
typical angina      46
Name: cp, dtype: int64
```



```
Fasting blood sugar distribution:
False    692
True     138
Name: fbs, dtype: int64

Resting electrocardiographic results distribution:
normal              551
lv hypertrophy      188
st-t abnormality    179
Name: restecg, dtype: int64

Exercise induced angina distribution:
False    528
True     337
Name: exang, dtype: int64

Number of major vessels distribution:
0.0    181
1.0     67
2.0     41
3.0     20
Name: ca, dtype: int64

Thalassemia distribution:
normal               196
reversable defect    192
fixed defect          46
Name: thal, dtype: int64
```

```
# Get statistical information for numeric columns
numeric_columns = train_data.select_dtypes(include='number')
statistics = numeric_columns.describe()

# Print the statistical information
print(statistics)
```

OUTPUT:



```
               id         age    trestbps        chol      thalch     oldpeak  \
count  920.000000  920.000000  861.000000  890.000000  865.000000  858.000000
mean   460.500000   53.510870  132.132404  199.130337  137.545665    0.878788
std    265.725422    9.424685   19.066070  110.780810   25.926276    1.091226
min      1.000000   28.000000    0.000000    0.000000   60.000000   -2.600000
25%    230.750000   47.000000  120.000000  175.000000  120.000000    0.000000
50%    460.500000   54.000000  130.000000  223.000000  140.000000    0.500000
75%    690.250000   60.000000  140.000000  268.000000  157.000000    1.500000
max    920.000000   77.000000  200.000000  603.000000  202.000000    6.200000

               ca         num
count  309.000000  920.000000
mean     0.676375    0.995652
std      0.935653    1.142693
min      0.000000    0.000000
25%      0.000000    0.000000
50%      0.000000    1.000000
75%      1.000000    2.000000
max      3.000000    4.000000
```

RESULT:

Public dataset is loaded and it is summarised.