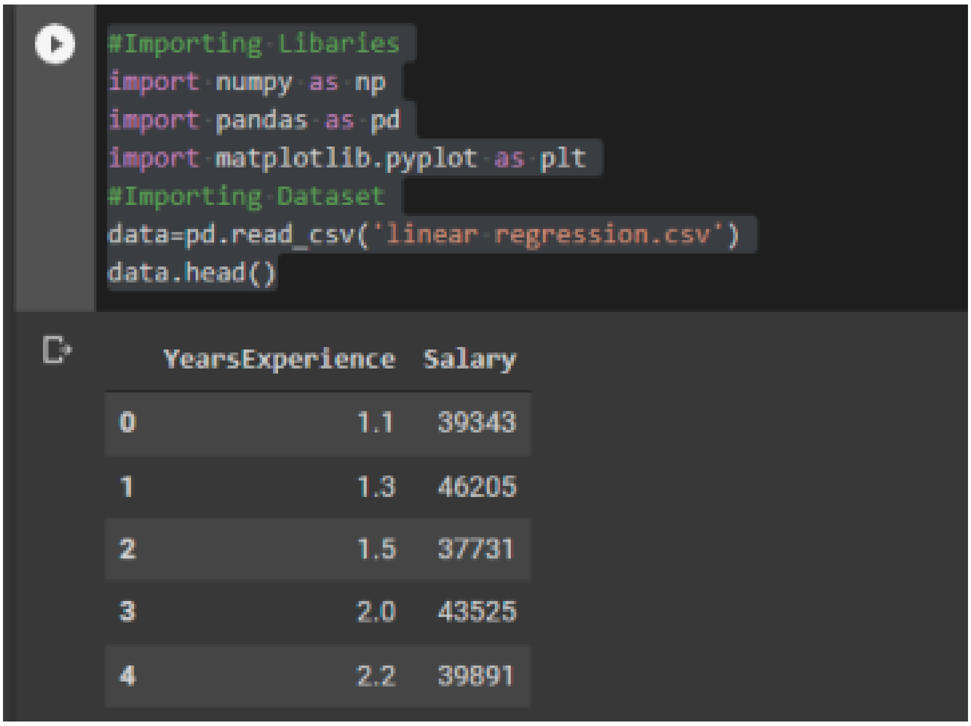
Name: Nilesh Kawar

Roll No. 59

Practical 2

Q1. Write a program to implement Linear Regression in python.
Program with Output:

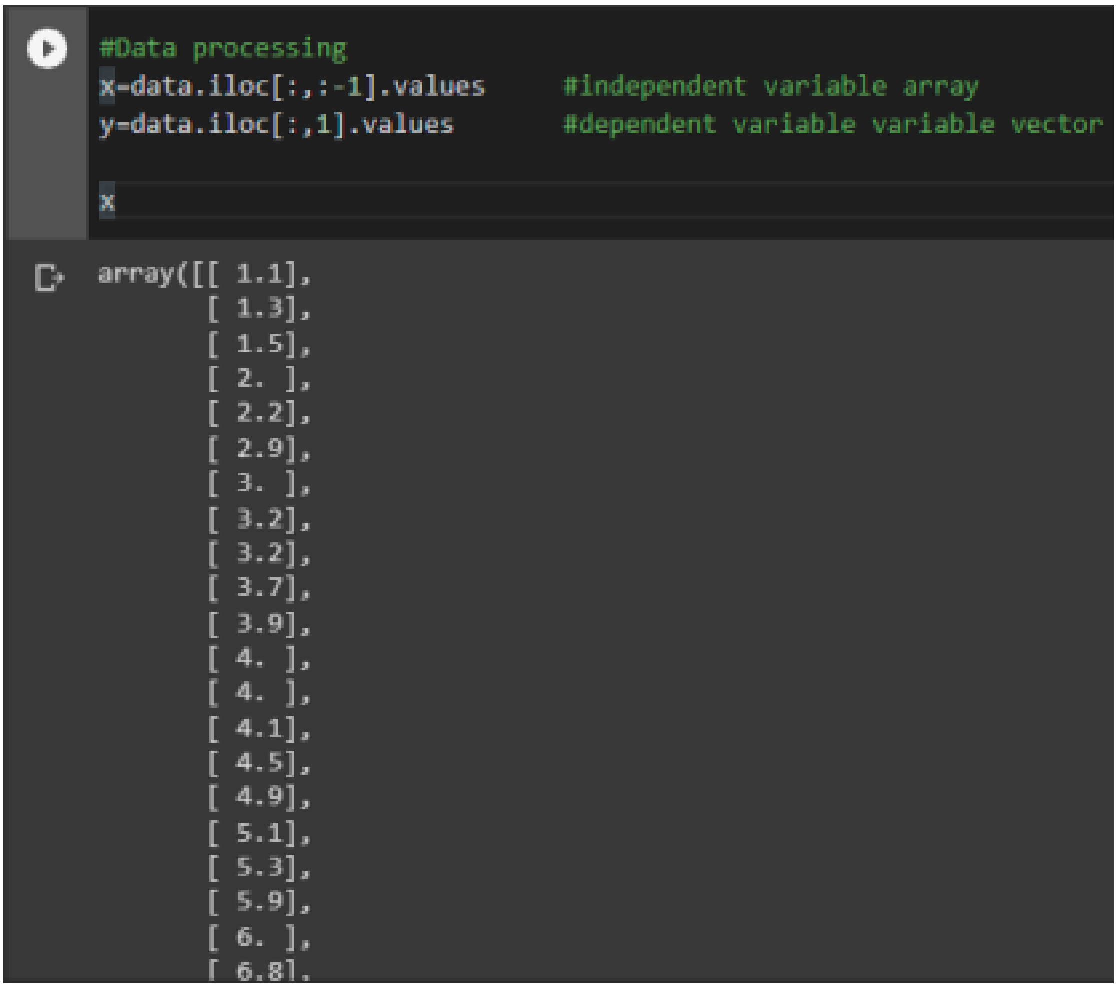
```
#Importing Libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#Importing Dataset
data=pd.read_csv('linear regression.csv')
data.head()
```



```
#Importing Libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#Importing Dataset
data=pd.read_csv('linear regression.csv')
data.head()
```

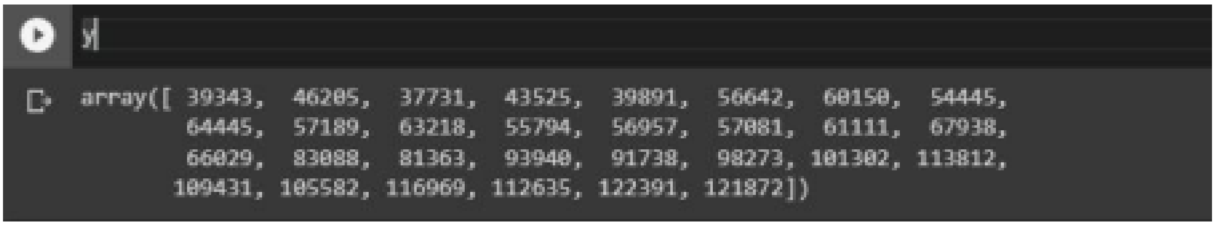
```
   YearsExperience  Salary
0              1.1   39343
1              1.3   46205
2              1.5   37731
3              2.0   43525
4              2.2   39891
```

```
#Data processing
x=data.iloc[:,:-1].values          #independent variable array
y=data.iloc[:,1].values            #dependent variable variable
                                   vector

x
```



```
#Data processing
x=data.iloc[:,:-1].values     #independent variable array
y=data.iloc[:,1].values       #dependent variable variable vector

x
```

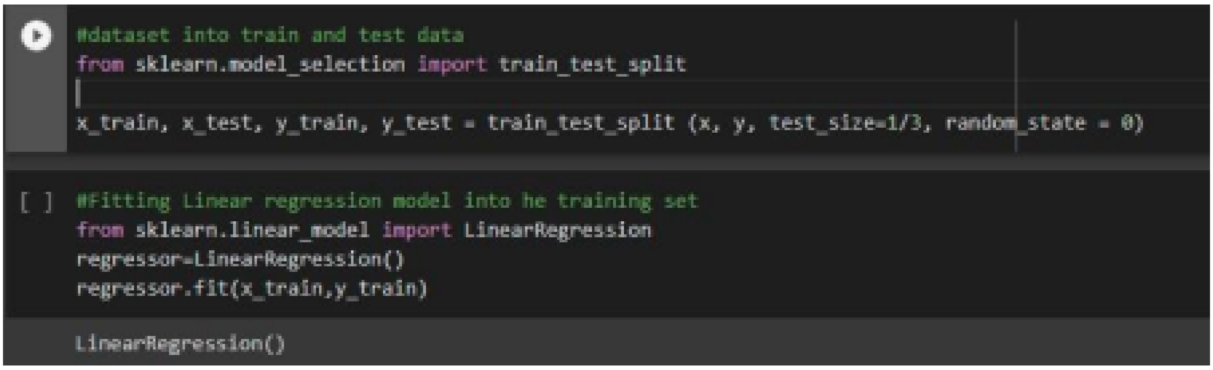
```
array([[ 1.1],
       [ 1.3],
       [ 1.5],
       [ 2. ],
       [ 2.2],
       [ 2.9],
       [ 3. ],
       [ 3.2],
       [ 3.2],
       [ 3.7],
       [ 3.9],
       [ 4. ],
       [ 4. ],
       [ 4.1],
       [ 4.5],
       [ 4.9],
       [ 5.1],
       [ 5.3],
       [ 5.9],
       [ 6. ],
       [ 6.8],
```



```
y
```

```
array([ 39343,  46205,  37731,  43525,  39891,  56642,  60150,  54445,
        64445,  57189,  63218,  55794,  56957,  57081,  61111,  67938,
        66029,  83088,  81363,  93940,  91738,  98273, 101302, 113812,
       109431, 105582, 116969, 112635, 122391, 121872])
```

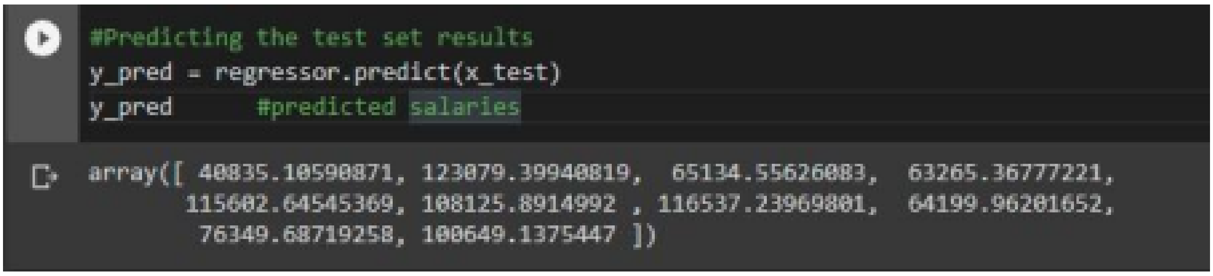
```
#dataset into train and test data
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split (x, y,
test_size=1/ 3, random_state = 0)




#Fitting Linear regression model into he training set
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(x_train,y_train)
```



```
#dataset into train and test data
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split (x, y, test_size=1/3, random_state = 0)
```

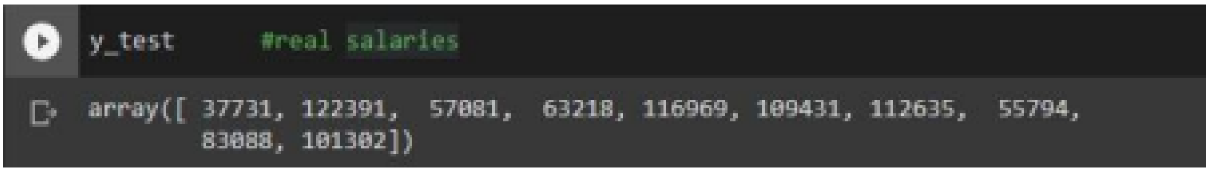
```
[ ] #Fitting Linear regression model into he training set
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(x_train,y_train)

LinearRegression()
```

```
#Predicting the test set results
y_pred = regressor.predict(x_test)
y_pred #predicted salaries
```



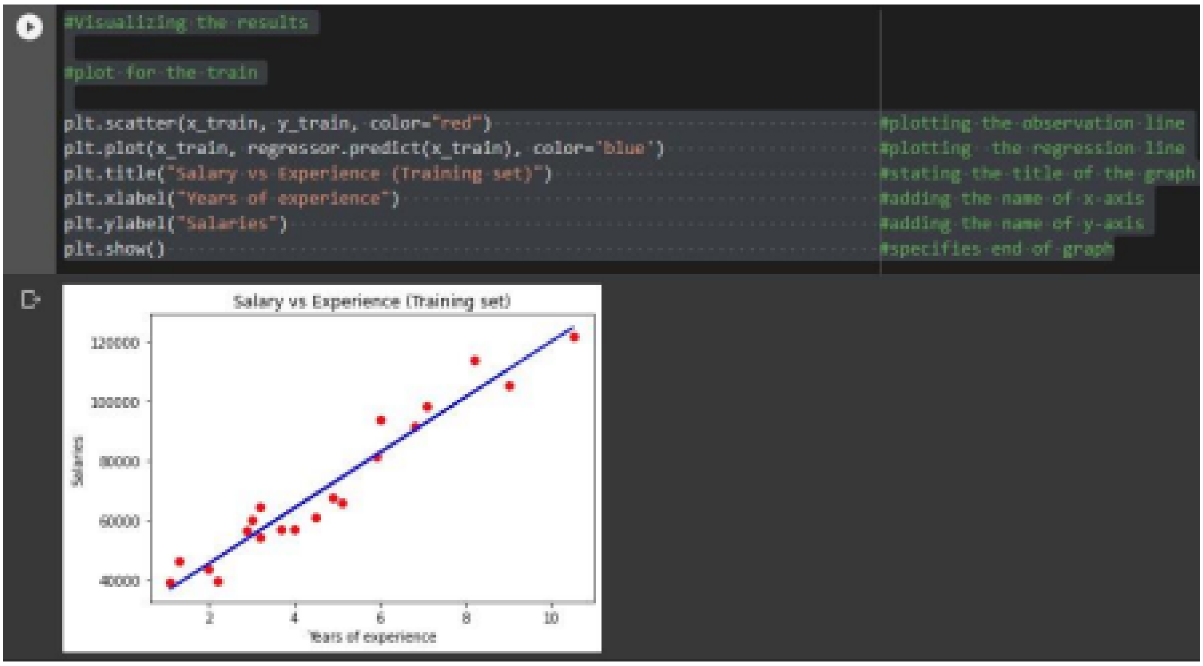
```
#Predicting the test set results
y_pred = regressor.predict(x_test)
y_pred      #predicted salaries

array([ 40835.10590871, 123079.39940819,  65134.55626083,  63265.36777221,
       115602.64545369, 108125.8914992 , 116537.23969801,  64199.96201652,
        76349.68719258, 100649.1375447 ])
```

```
y_test #real salaries
```



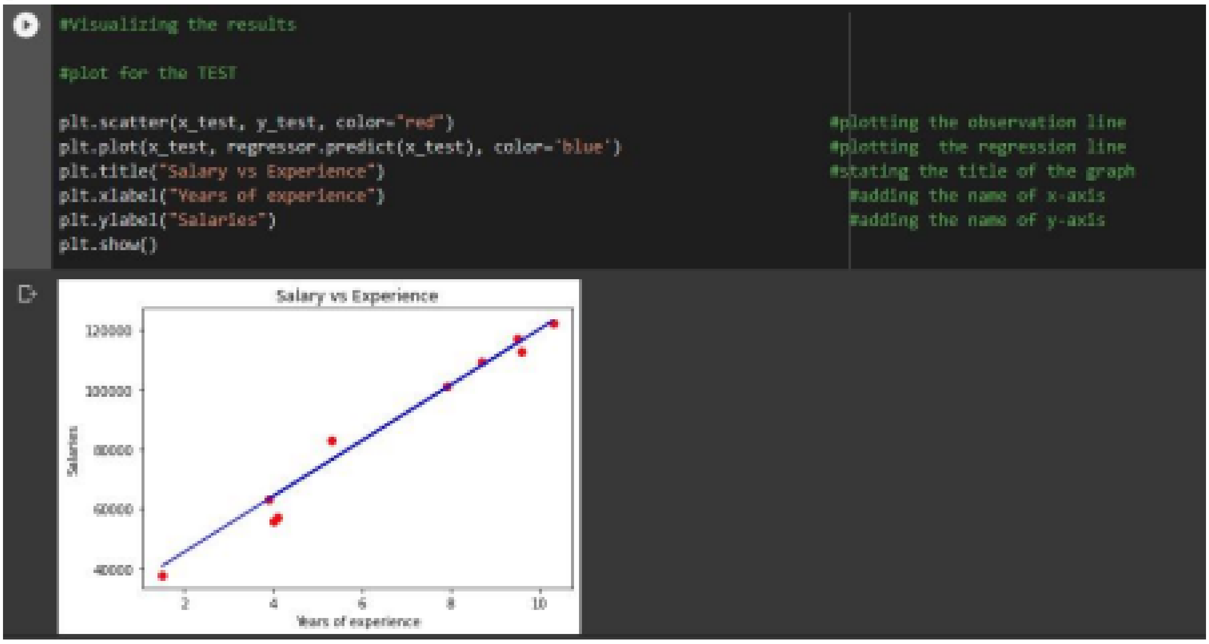
```
y_test      #real salaries

array([ 37731, 122391,  57081,  63218, 116969, 109431, 112635,  55794,
        83088, 101302])
```

```
#Visualizing the results

#plot for the train

plt.scatter(x_train, y_train, color="red")
         #plotting the observation line
plt.plot(x_train, regressor.predict(x_train), color='blue')
               #plotting the regression line
plt.title("Salary vs Experience (Training set)")
         #stating the title of the graph
plt.xlabel("Years of experience")
         #adding the name of x-axis
plt.ylabel("Salaries")
         #adding the name of y-axis
plt.show()
         #specifies end of graph
```



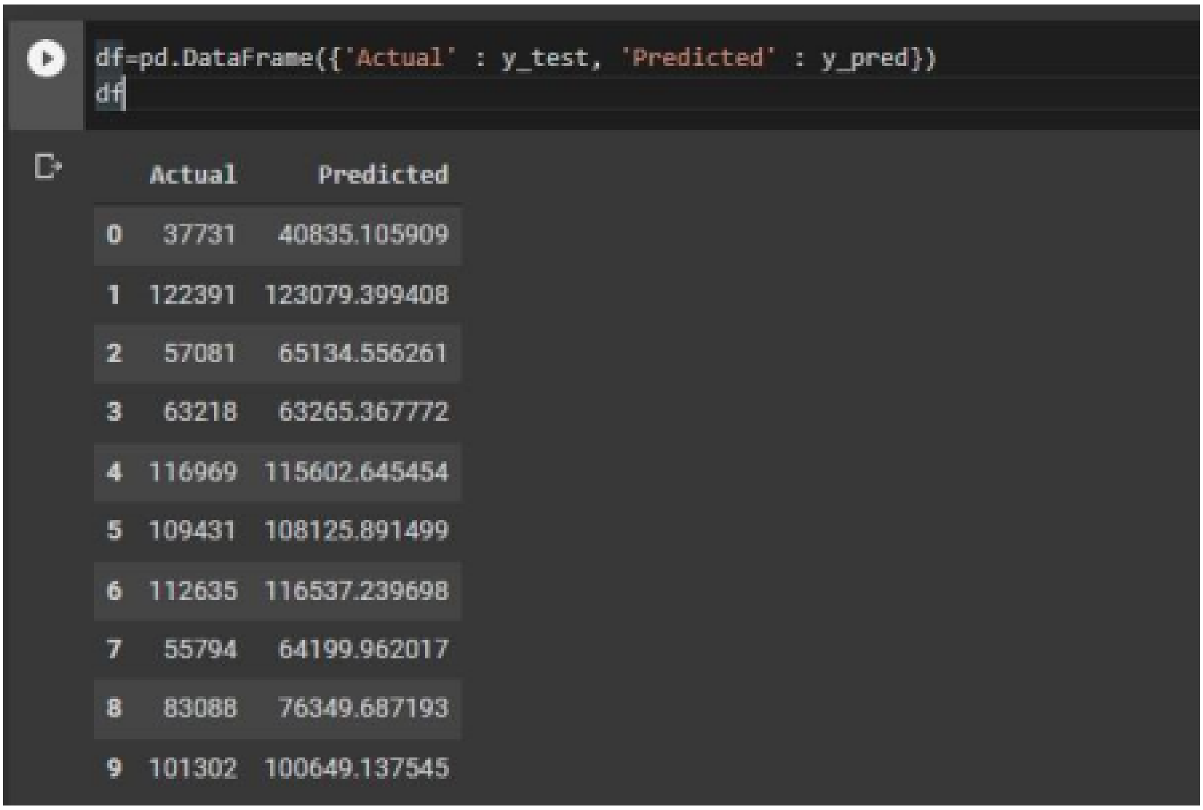
```
#Visualizing the results

#plot for the train

plt.scatter(x_train, y_train, color="red")                              #plotting the observation line
plt.plot(x_train, regressor.predict(x_train), color='blue')             #plotting  the regression line
plt.title("Salary vs Experience (Training set)")                        #stating the title of the graph
plt.xlabel("Years of experience")                                       #adding the name of x-axis
plt.ylabel("Salaries")                                                  #adding the name of y-axis
plt.show()                                                              #specifies end of graph
```

```
#Visualizing the results

#plot for the TEST

plt.scatter(x_test, y_test, color="red")
       #plotting the observation line
plt.plot(x_test, regressor.predict(x_test), color='blue')
              #plotting the regression line
plt.title("Salary vs Experience")
       #stating the title of the graph
plt.xlabel("Years of experience")
         #adding the name of x-axis
plt.ylabel("Salaries")
         #adding the name of y-axis
plt.show()
```



```
#Visualizing the results

#plot for the TEST

plt.scatter(x_test, y_test, color="red")                          #plotting the observation line
plt.plot(x_test, regressor.predict(x_test), color='blue')         #plotting  the regression line
plt.title("Salary vs Experience")                                 #stating the title of the graph
plt.xlabel("Years of experience")                                   #adding the name of x-axis
plt.ylabel("Salaries")                                              #adding the name of y-axis
plt.show()
```

```
df=pd.DataFrame({'Actual' : y_test, 'Predicted' :
y_pred}) df
```



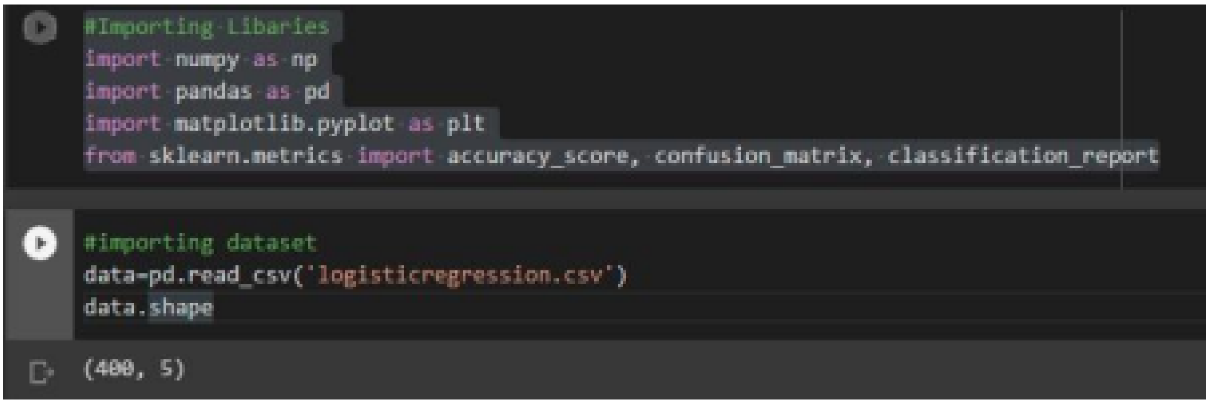
```
df=pd.DataFrame({'Actual' : y_test, 'Predicted' : y_pred})
df
```

```
   Actual     Predicted
0   37731   40835.105909
1  122391  123079.399408
2   57081   65134.556261
3   63218   63265.367772
4  116969  115602.645454
5  109431  108125.891499
6  112635  116537.239698
7   55794   64199.962017
8   83088   76349.687193
9  101302  100649.137545
```

Q2. Write a program to implement Logistic Regression in python

Program with Output:

```
#Importing Libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, confusion_matrix,
classific ation_report

#importing dataset
data=pd.read_csv('logisticregression.csv')
data.shape
```



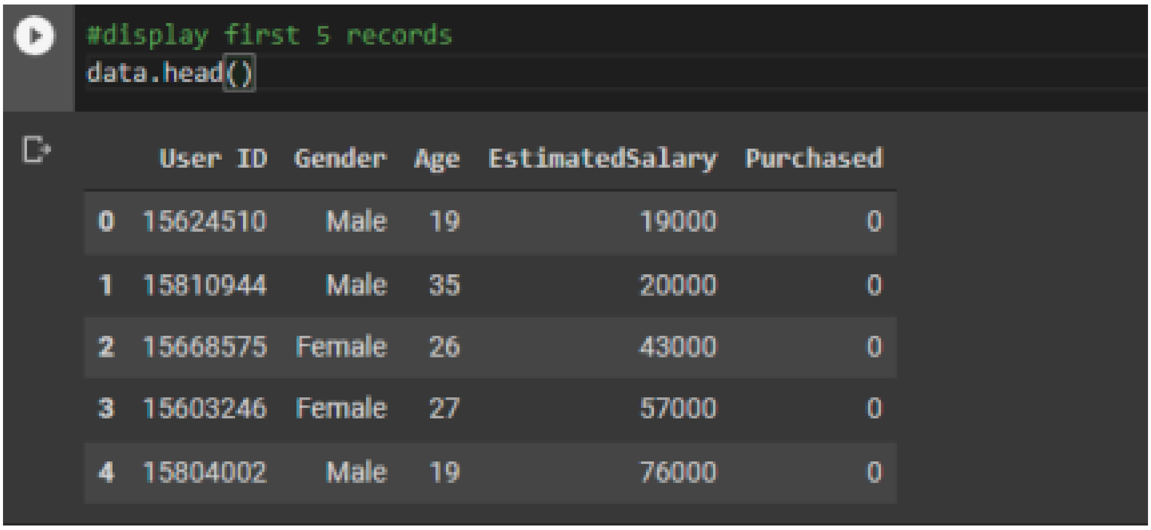
```
#Importing Libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
#importing dataset
data=pd.read_csv('logisticregression.csv')
data.shape
```

```
(400, 5)
```

```
#display first 5 records
```

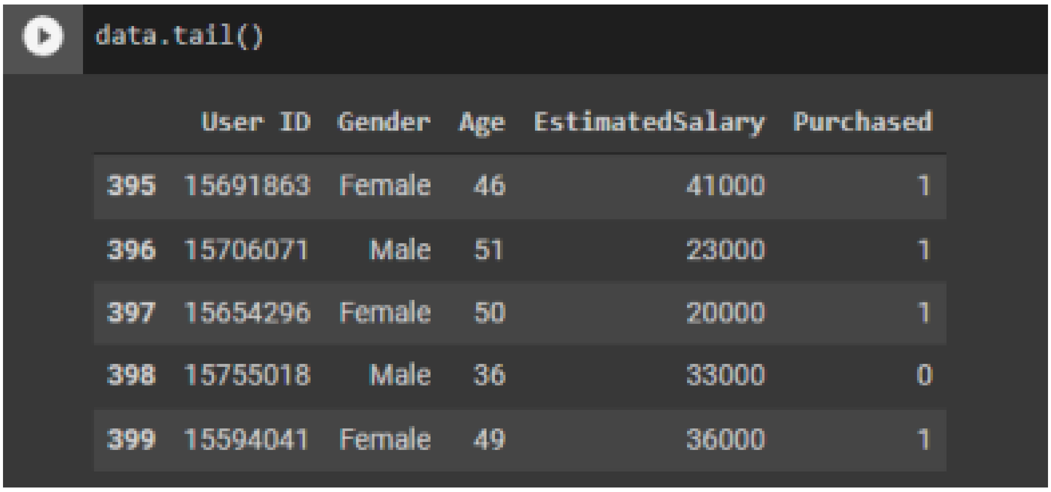
```
data.head()
```



```
#display first 5 records
data.head()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

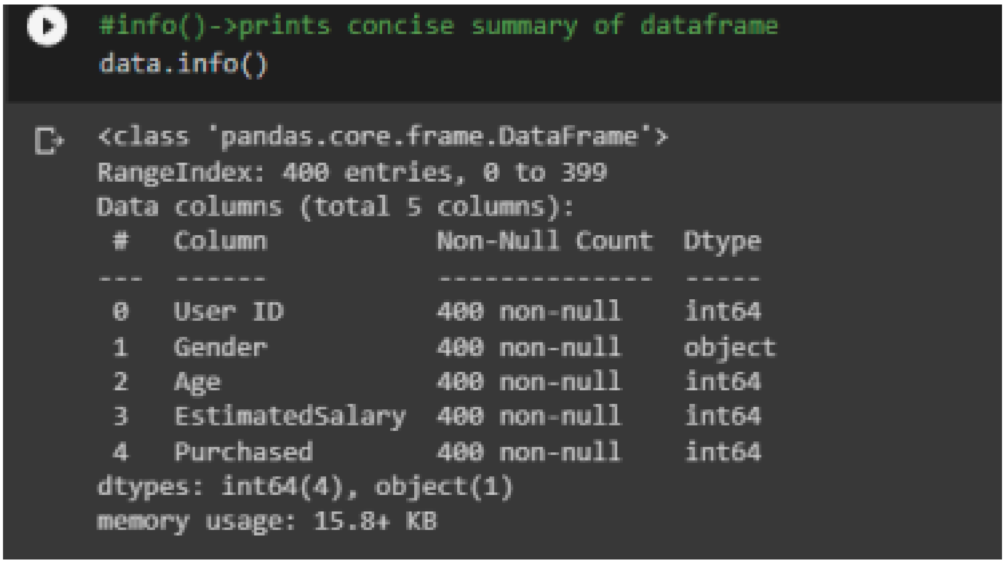
data.tail()



```
data.tail()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

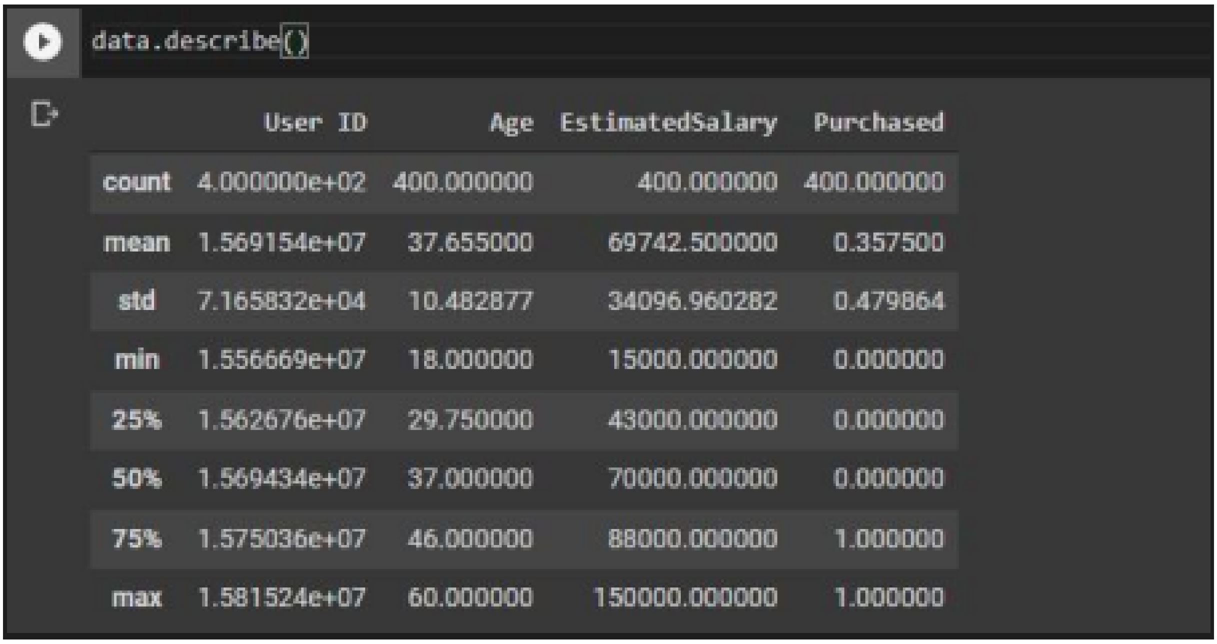
```
#info()->prints concise summary of dataframe
data.info()
```



```
#info()->prints concise summary of dataframe
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   User ID          400 non-null    int64
 1   Gender           400 non-null    object
 2   Age              400 non-null    int64
 3   EstimatedSalary  400 non-null    int64
 4   Purchased        400 non-null    int64
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

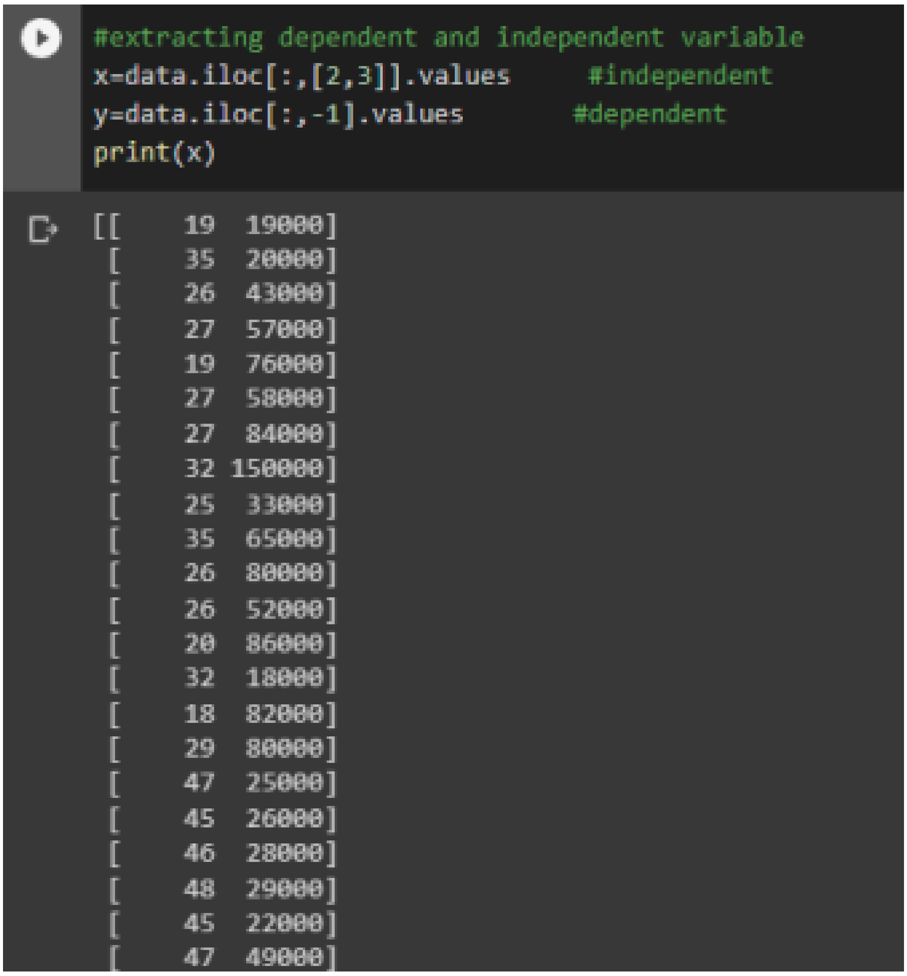
data.describe()



```
data.describe()
```

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864
min	1.556669e+07	18.000000	15000.000000	0.000000
25%	1.562676e+07	29.750000	43000.000000	0.000000
50%	1.569434e+07	37.000000	70000.000000	0.000000
75%	1.575036e+07	46.000000	88000.000000	1.000000
max	1.581524e+07	60.000000	150000.000000	1.000000

```
#extracting dependent and independent variable
x=data.iloc[:,[2,3]].values #independent
y=data.iloc[:,-1].values #dependent
print(x)
```



```
#extracting dependent and independent variable
x=data.iloc[:,[2,3]].values     #independent
y=data.iloc[:,-1].values       #dependent
print(x)
```

```
[[    19  19000]
 [    35  20000]
 [    26  43000]
 [    27  57000]
 [    19  76000]
 [    27  58000]
 [    27  84000]
 [    32 150000]
 [    25  33000]
 [    35  65000]
 [    26  80000]
 [    26  52000]
 [    20  86000]
 [    32  18000]
 [    18  82000]
 [    29  80000]
 [    47  25000]
 [    45  26000]
 [    46  28000]
 [    48  29000]
 [    45  22000]
 [    47  49000]
```

```
print(y)
```

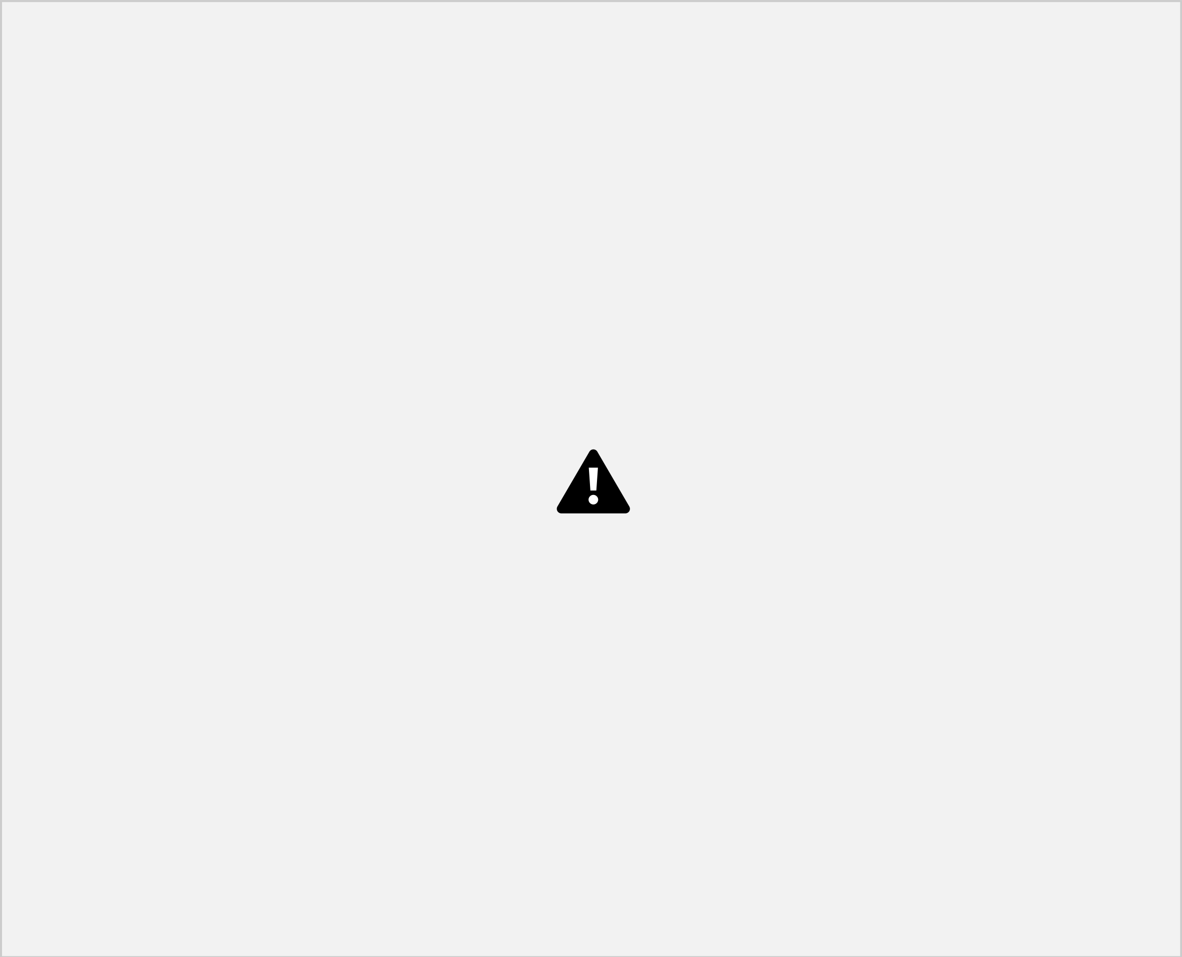
```
#splitting the dataset into training and test set
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test =
train_test_split(x,y,test_size=0.30, random_state=0)
```

```
x_test
```

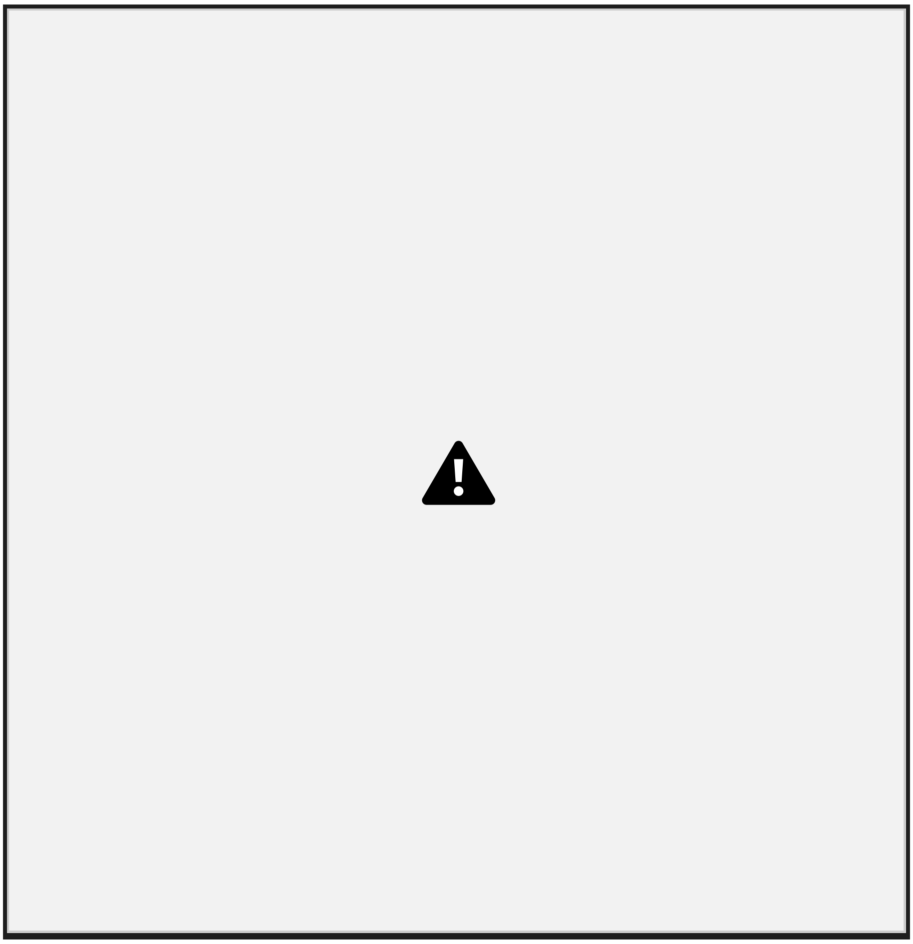
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
#feature Scaling
from sklearn.preprocessing import StandardScaler
st_x=StandardScaler()
x_train=st_x.fit_transform(x_train)
x_test=st_x.fit_transform(x_test)
```

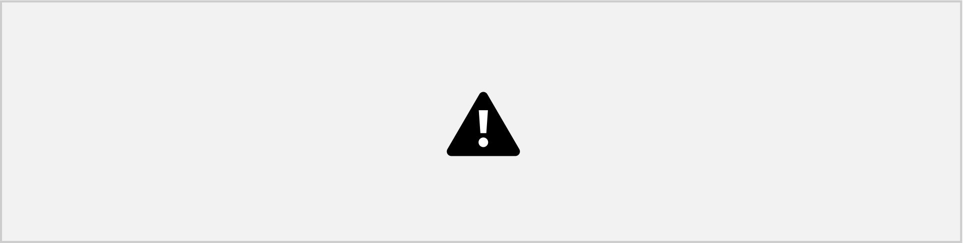
```
print(x_train)
```



```
print(x_test)
```



```
#fitting logistic regression model to thr training
set from sklearn.linear_model import
LogisticRegression
Classifier=LogisticRegression(random_state=0)
Classifier.fit(x_train,y_train)
```



```
#predicting the test set result
y_pred=Classifier.predict(x_test)
y_trainPred=Classifier.predict(x_train)
print(y_pred)
print(y_trainPred)
```

```
#check model score
Classifier.score(x_test,y_test)
```

```
#creating the confusion matrix
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
cm
```

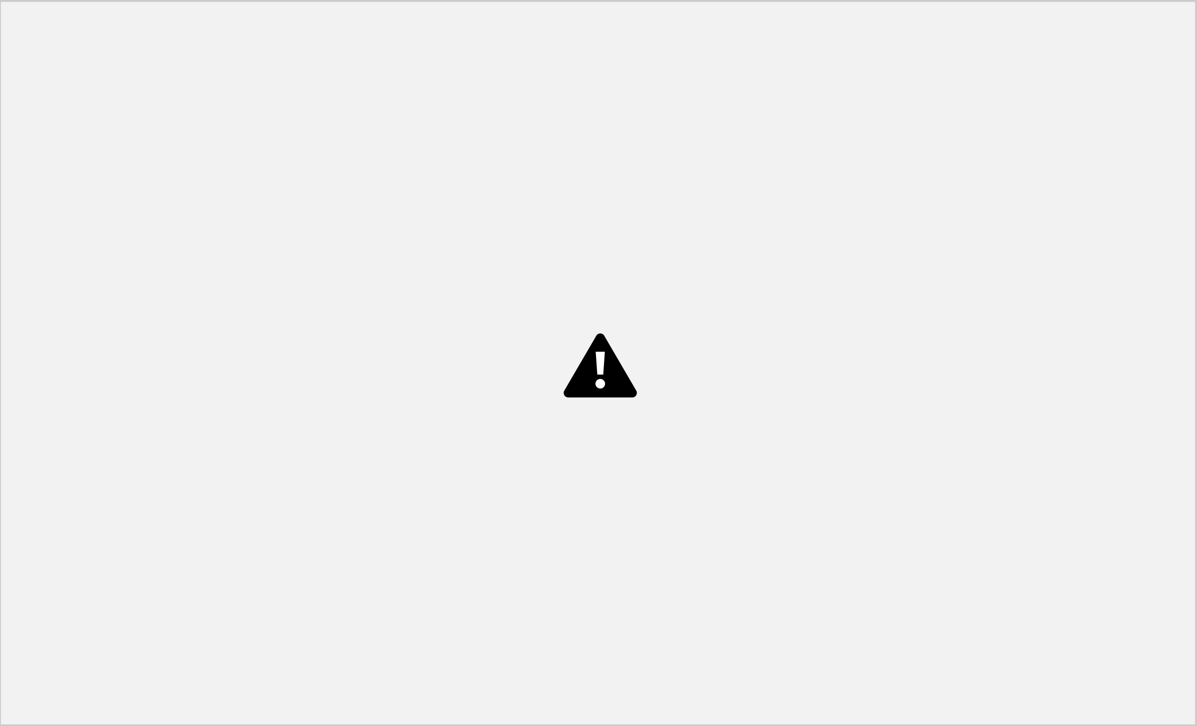
```
#confusion matrix
print(confusion_matrix(y_train,y_trainPred))
```

```
#creating classification report
print(classification_report(y_test, y_pred))
```

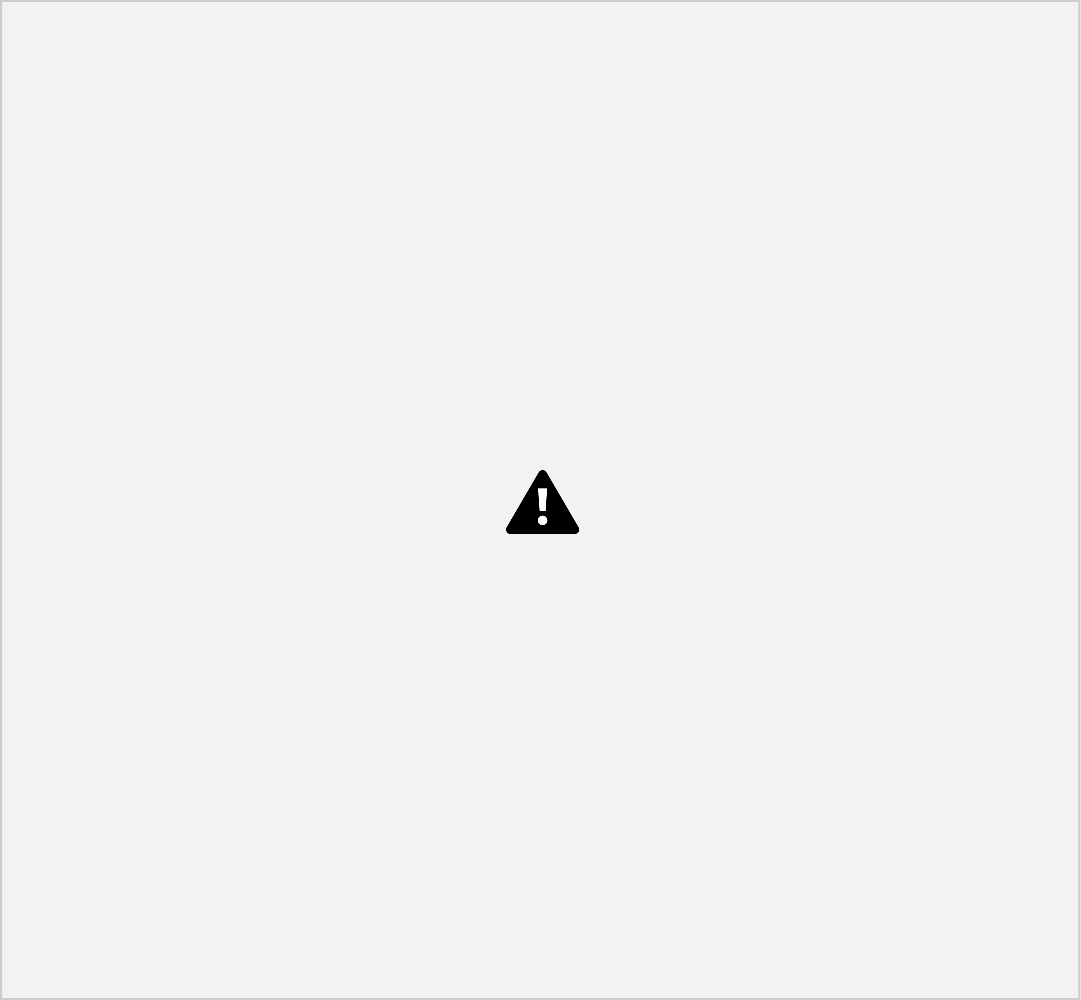
Q3. Write a python code to implement K-Nearest Neighbor(KNN) Algorithm for Machine Learning

Program with Output:

```
#import Libraries
import numpy as np #perform a wide variety of mathematical operations
o n
import pandas as pd
from sklearn.metrics import confusion_matrix, classification_report

#importing dataset
data=pd.read_csv('logisticregression.csv')
data.head()
```



```
#extracting dependent and independent variable
x=data.iloc[:,[2,3]].values #independent variable array
y=data.iloc[:,4].values #dependent variable vector print(x)
```



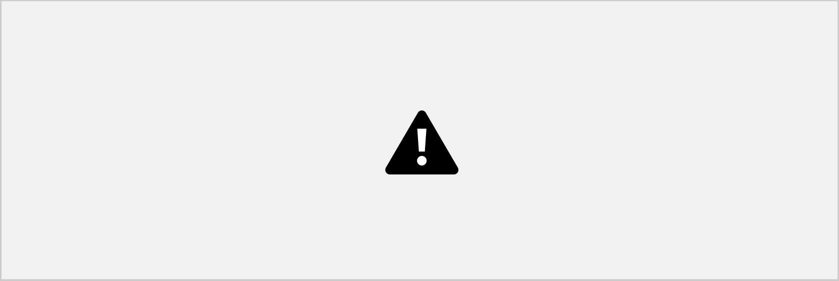
```
print(y)
```

```
#splitting the dataset into training and testing set
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test =
train_test_split(x,y,test_size=0.30, random_state=0)


#feature Scaling
from sklearn.preprocessing import StandardScaler
st_x=StandardScaler()
x_train=st_x.fit_transform(x_train)
x_test=st_x.transform(x_test)


#fitting K-NN classifier to the training set
```

```
from sklearn.neighbors import KNeighborsClassifier
Classifier=KNeighborsClassifier(n_neighbors=5, metric='minkowski',
p=2) Classifier.fit(x_train,y_train)
```

```
y_pred=Classifier.predict(x_test)
print(y_pred)
```

```
#creating the confusion matrix
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
cm
```



```
print(classification_report(y_test,y_pred))
```