aicte-project2-phase2-1

November 20, 2023

1 import the Required libraries

[1]:
```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
```

2 Loading the Train Data

[10]:
```
train=pd.read_html("https://docs.google.com/spreadsheets/d/e/
↪2PACX-1vRTK2NvcndgPX41Czu6Ft2Ho_nE-z50BgTqdzwFW0rsJ2nvyNLe2DoIg1COzUbgw80oaRBjfy5-WtFk/
↪pubhtml")
train
```

[10]:
```
[     Unnamed: 0 Unnamed: 1   Unnamed: 2
 0             1          x             y
 1             2         24  21.54945196
 2             3         50  47.46446305
 3             4         15  17.21865634
 4             5         38  36.58639803
 ..          …         …          …
 696         697         58  58.59500642
 697         698         93  94.62509374
 698         699         82  88.60376995
 699         700         66  63.64868529
 700         701         97   94.9752655

 [701 rows x 3 columns]]
```

[3]:
```
train_set=train[0]
train_set.to_csv('train.csv',index=False)
```

[4]:
```
train=pd.read_csv("train.csv")
```

[5]:
```
train
```

[5]:
```
     Unnamed: 0 Unnamed: 1   Unnamed: 2
0             1          x            y
1             2         24  21.54945196
2             3         50  47.46446305
3             4         15  17.21865634
4             5         38  36.58639803
..          …         …          …
696         697         58  58.59500642
697         698         93  94.62509374
698         699         82  88.60376995
699         700         66  63.64868529
700         701         97   94.9752655

[701 rows x 3 columns]
```

3 Making the train data more concise

[6]:
```
trains=train.drop("Unnamed: 0",axis=1)
trains.columns=trains.iloc[0]
train_data=trains.drop(0)
train_data
```

[6]:
```
0     x            y
1    24  21.54945196
2    50  47.46446305
3    15  17.21865634
4    38  36.58639803
5    87  87.28898389
..   ..          …
696  58  58.59500642
697  93  94.62509374
698  82  88.60376995
699  66  63.64868529
700  97   94.9752655

[700 rows x 2 columns]
```

4 Loading the test data

[11]:
```
test=pd.read_html("https://docs.google.com/spreadsheets/d/e/
  ↪2PACX-1vRyvZ7lknwiSghK9aen1SaTEYoN3JS40rrGLpcyrsVZy1tB2T4gn6Y3-cdzPUFCPMmmqREWefW3kl4_/
  ↪pubhtml")
test
```

[11]:
```
[     Unnamed: 0 Unnamed: 1   Unnamed: 2
 0             1          x            y
 1             2         77  79.77515201
 2             3         21  23.17727887
 3             4         22  25.60926156
 4             5         20  17.85738813
 ..          …        …          …
 296         297         71   68.5458879
 297         298         46  47.33487629
 298         299         55  54.09063686
 299         300         62  63.29717058
 300         301         47  52.45946688

 [301 rows x 3 columns]]
```

[12]:
```
testset=test[0]
testset.to_csv('test.csv',index=False)
```

[13]:
```
test=pd.read_csv("test.csv")
test
```

[13]:
```
     Unnamed: 0 Unnamed: 1   Unnamed: 2
0             1          x            y
1             2         77  79.77515201
2             3         21  23.17727887
3             4         22  25.60926156
4             5         20  17.85738813
..          …        …          …
296         297         71   68.5458879
297         298         46  47.33487629
298         299         55  54.09063686
299         300         62  63.29717058
300         301         47  52.45946688

[301 rows x 3 columns]
```

[ ]:
```
#making the test data more concise
```

[14]:
```
tests=test.drop("Unnamed: 0",axis=1)
tests.columns=tests.iloc[0]
test_data=tests.drop(0)
test_data
```

[14]:
```
0     x            y
1    77  79.77515201
2    21  23.17727887
3    22  25.60926156
```

```
4    20  17.85738813
5    36  41.84986439
..   ..          …
296  71   68.5458879
297  46  47.33487629
298  55  54.09063686
299  62  63.29717058
300  47  52.45946688

[300 rows x 2 columns]
```

5 Droping the null rows

[15]:
```
train_dataset=train_data.dropna(subset=['y'])
train_dataset
```

[15]:
```
0     x           y
1    24  21.54945196
2    50  47.46446305
3    15  17.21865634
4    38  36.58639803
5    87  87.28898389
..   ..          …
696  58  58.59500642
697  93  94.62509374
698  82  88.60376995
699  66  63.64868529
700  97   94.9752655

[699 rows x 2 columns]
```

6 Distinguishing X_train and y_train

[17]:
```
x_train=train_dataset[['x']].astype(float)
y_train=train_dataset[['y']].astype(float)
```

[18]:
```
x_train
```

[18]:
```
0       x
1    24.0
2    50.0
3    15.0
4    38.0
5    87.0
..    …
```

```
696  58.0
697  93.0
698  82.0
699  66.0
700  97.0

[699 rows x 1 columns]
```

[19]:
```
y_train
```

[19]:
```
0           y
1    21.549452
2    47.464463
3    17.218656
4    36.586398
5    87.288984
..         …
696  58.595006
697  94.625094
698  88.603770
699  63.648685
700  94.975266

[699 rows x 1 columns]
```

7 Distinguishing x_test and y_test

[25]:
```
x_test=test_data[['x']].astype(float)
y_test=test_data[['y']].astype(float)
```

[26]:
```
x_test
```

[26]:
```
0       x
1    77.0
2    21.0
3    22.0
4    20.0
5    36.0
..    …
296  71.0
297  46.0
298  55.0
299  62.0
300  47.0

[300 rows x 1 columns]
```

[28]:
```
y_test
```

[28]:
```
0            y
1    79.775152
2    23.177279
3    25.609262
4    17.857388
5    41.849864
..         …
296  68.545888
297  47.334876
298  54.090637
299  63.297171
300  52.459467

[300 rows x 1 columns]
```

8 Linear Regression model for train

[29]:
```
model=LinearRegression()
model.fit(x_train,y_train)
```

[29]:
```
LinearRegression()
```

9 predicting the test data

[30]:
```
y_pred=model.predict(x_test)
y_pred
```

[30]:
```
array([[76.94327594],
       [20.90651855],
       [21.90717494],
       [19.90586217],
       [35.91636428],
       [14.90258026],
       [61.93343021],
       [94.95509081],
       [19.90586217],
       [ 4.89601644],
       [ 3.89536006],
       [18.90520579],
       [95.95574719],
       [61.93343021],
       [35.91636428],
       [14.90258026],
```

```
[64.93539936],
[13.90192388],
[86.94983976],
[68.93802488],
[88.95115252],
[50.92621001],
[88.95115252],
[26.91045685],
[96.95640358],
[57.93080468],
[78.9445887 ],
[20.90651855],
[92.95377805],
[26.91045685],
[98.95771634],
[30.91308237],
[32.91439514],
[79.94524508],
[27.91111323],
[46.92358448],
[52.92752277],
[68.93802488],
[27.91111323],
[32.91439514],
[90.95246528],
[70.93933765],
[49.92555363],
[75.94261956],
[ 3.89536006],
[36.91702066],
[69.93868127],
[67.9373685 ],
[39.91898981],
[34.9157079 ],
[93.95443443],
[87.95049614],
[51.92686639],
[30.91308237],
[58.93146107],
[-0.10726546],
[38.91833343],
[63.93474297],
[68.93802488],
[56.9301483 ],
[12.9012675 ],
[71.93999403],
[75.94261956],
```

```
[60.93277383],
[81.94655785],
[17.90454941],
[40.91964619],
[49.92555363],
[54.92883554],
[12.9012675 ],
[45.9229281 ],
[12.9012675 ],
[78.9445887 ],
[52.92752277],
[14.90258026],
[27.91111323],
[80.94590147],
[68.93802488],
[51.92686639],
[83.94787061],
[67.9373685 ],
[26.91045685],
[55.92949192],
[47.92424086],
[39.91898981],
[38.91833343],
[81.94655785],
[99.95837272],
[58.93146107],
[42.92095896],
[66.93671212],
[37.91767705],
[62.93408659],
[90.95246528],
[59.93211745],
[13.90192388],
[20.90651855],
[86.94983976],
[72.94065041],
[31.91373876],
[ 1.8940473 ],
[81.94655785],
[18.90520579],
[73.94130679],
[41.92030257],
[11.90061112],
[ 0.89339092],
[89.9518089 ],
[88.95115252],
[-0.10726546],
```

```
[40.91964619],
[15.90323665],
[93.95443443],
[96.95640358],
[65.93605574],
[23.9084877 ],
[16.90389303],
[89.9518089 ],
[12.9012675 ],
[-0.10726546],
[63.93474297],
[95.95574719],
[97.95705996],
[11.90061112],
[40.91964619],
[46.92358448],
[77.94393232],
[19.90586217],
[88.95115252],
[28.91176961],
[63.93474297],
[74.94196317],
[11.90061112],
[24.90914408],
[27.91111323],
[29.91242599],
[64.93539936],
[58.93146107],
[63.93474297],
[52.92752277],
[70.93933765],
[96.95640358],
[72.94065041],
[ 8.89864197],
[11.90061112],
[62.93408659],
[98.95771634],
[59.93211745],
[34.9157079 ],
[ 1.8940473 ],
[59.93211745],
[31.91373876],
[93.95443443],
[83.94787061],
[62.93408659],
[21.90717494],
[80.94590147],
```

```
[92.95377805],
[32.91439514],
[ 6.89732921],
[41.92030257],
[45.9229281 ],
[53.92817916],
[15.90323665],
[48.92489725],
[42.92095896],
[94.95509081],
[65.93605574],
[20.90651855],
[34.9157079 ],
[79.94524508],
[36.91702066],
[53.92817916],
[55.92949192],
[ 0.89339092],
[31.91373876],
[57.93080468],
[31.91373876],
[45.9229281 ],
[71.93999403],
[16.90389303],
[96.95640358],
[92.95377805],
[90.95246528],
[36.91702066],
[ 3.89536006],
[53.92817916],
[50.92621001],
[26.91045685],
[45.9229281 ],
[91.95312167],
[72.94065041],
[76.94327594],
[90.95246528],
[60.93277383],
[98.95771634],
[ 3.89536006],
[71.93999403],
[18.90520579],
[56.9301483 ],
[77.94393232],
[25.90980046],
[73.94130679],
[89.9518089 ],
```

```
[65.93605574],
[12.9012675 ],
[39.91898981],
[76.94327594],
[66.93671212],
[74.94196317],
[22.90783132],
[44.92227172],
[58.93146107],
[43.92161534],
[22.90783132],
[54.92883554],
[54.92883554],
[94.95509081],
[11.90061112],
[ 3.89536006],
[ 6.89732921],
[99.95837272],
[47.92424086],
[41.92030257],
[95.95574719],
[38.91833343],
[99.95837272],
[86.94983976],
[13.90192388],
[13.90192388],
[36.91702066],
[ 4.89601644],
[87.95049614],
[90.95246528],
[64.93539936],
[73.94130679],
[55.92949192],
[15.90323665],
[ 4.89601644],
[27.91111323],
[91.95312167],
[45.9229281 ],
[53.92817916],
[38.91833343],
[43.92161534],
[30.91308237],
[67.9373685 ],
[85.94918338],
[89.9518089 ],
[37.91767705],
[20.90651855],
```

```
[94.95509081],
[55.92949192],
[59.93211745],
[64.93539936],
[77.94393232],
[88.95115252],
[ 5.89667283],
[66.93671212],
[35.91636428],
[15.90323665],
[99.95837272],
[44.92227172],
[72.94065041],
[56.9301483 ],
[19.90586217],
[75.94261956],
[33.91505152],
[54.92883554],
[71.93999403],
[54.92883554],
[ 7.89798559],
[55.92949192],
[71.93999403],
[57.93080468],
[ 5.89667283],
[95.95574719],
[22.90783132],
[57.93080468],
[22.90783132],
[18.90520579],
[24.90914408],
[63.93474297],
[20.90651855],
[58.93146107],
[18.90520579],
[15.90323665],
[41.92030257],
[42.92095896],
[60.93277383],
[91.95312167],
[10.89995474],
[40.91964619],
[ 0.89339092],
[ 7.89798559],
[70.93933765],
[45.9229281 ],
[54.92883554],
```

```
       [61.93343021],
       [46.92358448]])
```

10 converting arrays to 1D

[32]:
```
y_test=y_test.squeeze()
y_pred=y_pred.squeeze()
```

11 Prediction

[34]:
```
result=pd.DataFrame({'Actual value':y_test,'predicted_value':y_pred})
result
```

[34]:

	Actual value	predicted_value
1	79.775152	76.943276
2	23.177279	20.906519
3	25.609262	21.907175
4	17.857388	19.905862
5	41.849864	35.916364
..	…	…
296	68.545888	70.939338
297	47.334876	45.922928
298	54.090637	54.928836
299	63.297171	61.933430
300	52.459467	46.923584

[300 rows x 2 columns]

12 Evaluating the model using mean_squared_error and r2_score

[36]:
```
mse=mean_squared_error(y_test,y_pred)
mse
```

[36]: 9.432922192039305

[37]:
```
r2=r2_score(y_test,y_pred)
r2
```

[37]: 0.9888014444327563

13 conclusion

14 A Lower MSE indicates that the models prediction are closer to the actual values,which suggests better predictive accuracy our R2 approximately equal to 1 means that the model perfectly explains all the variance in the target variable.