MUDSS Workshop-Copy1

October 15, 2021

1 IMPORT LIBRARIES

[11]:

```
import pandas as pd
import numpy as py
```

2 IMPORT DATASET

[18]:

```
df=pd.read_csv(r"C:\Users\mahru\Downloads\Copy .csv")
```

[19]:

```
df.head()
```

[19]:

```
      AT      V       AP     RH      PE
0  14.96  41.76  1024.07  73.17  463.26
1  25.18  62.96  1020.04  59.08  444.37
2   5.11  39.40  1012.16  92.14  488.56
3  20.86  57.32  1010.24  76.64  446.48
4  10.82  37.50  1009.23  96.62  473.90
```

3 Define x and y (independent and dependent variable)

[20]:

```
x=df.drop(['PE'],axis=1)
y=df['PE'].values
```

[21]:

```
print(x)
```

```
         AT      V       AP     RH
0     14.96  41.76  1024.07  73.17
1     25.18  62.96  1020.04  59.08
2      5.11  39.40  1012.16  92.14
3     20.86  57.32  1010.24  76.64
4     10.82  37.50  1009.23  96.62
…       …      …       …      …
9563  16.65  49.69  1014.01  91.00
9564  13.19  39.18  1023.67  66.78
9565  31.32  74.33  1012.92  36.48
9566  24.48  69.45  1013.86  62.39
9567  21.60  62.52  1017.23  67.87
```

```
[9568 rows x 4 columns]
```

```
[22]: print(y)
```

```
[463.26 444.37 488.56 … 429.57 435.74 453.28]
```

4 Split the dataset in training set and test set

```
[23]: from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

5 Train the model on the training set

```
[27]: from sklearn.linear_model import LinearRegression
      ml=LinearRegression()
      ml.fit(x_train,y_train)
```

```
[27]: LinearRegression()
```

6 Predict the test set results

```
[28]: y_pred=ml.predict(x_test)
      print(y_pred)
```

```
[431.40245096 458.61474119 462.81967423 … 432.47380825 436.16417243
 439.00714594]
```

```
[29]: ml.predict([[14.96,41.76,1024.07,73.17]])
```

```
[29]: array([467.34820092])
```

7 Evaluate the model

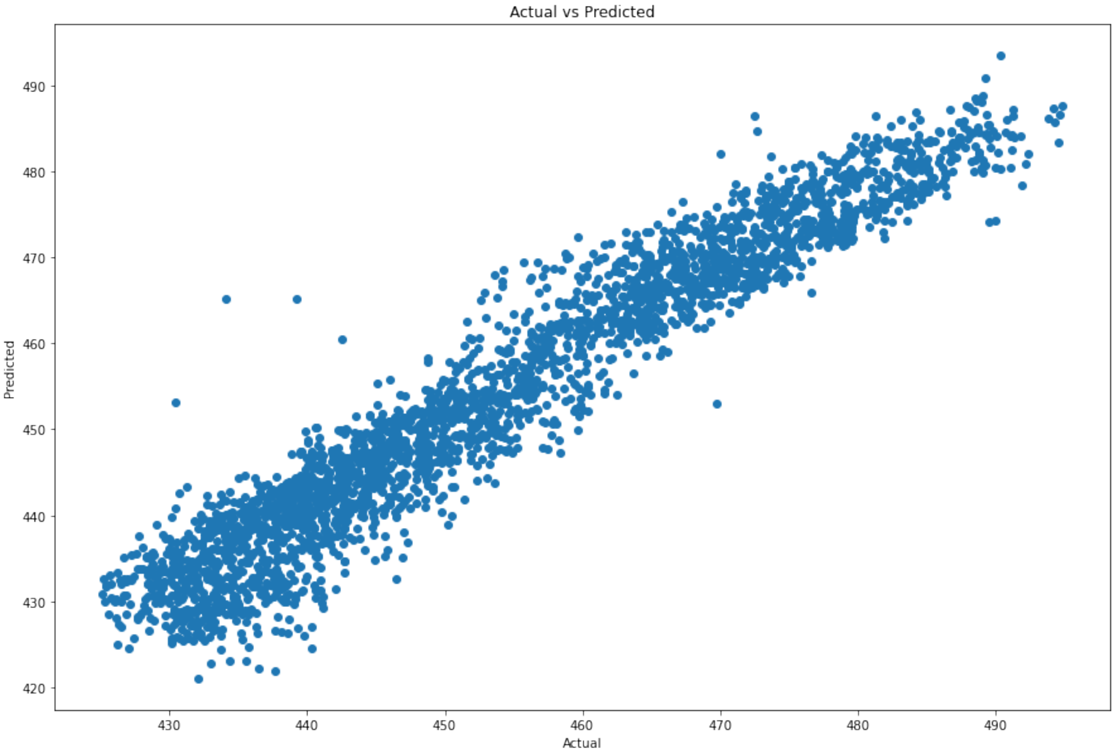
```
[30]: from sklearn.metrics import r2_score
      r2_score(y_test,y_pred)
```

```
[30]: 0.9304112159477682
```

8 Plot the results

```
[34]: import matplotlib.pyplot as plt
      plt.figure(figsize=(15,10))
      plt.scatter(y_test,y_pred)
      plt.xlabel('Actual')
      plt.ylabel('Predicted')
plt.title('Actual vs Predicted')
```

[34]:
```
Text(0.5, 1.0, 'Actual vs Predicted')
```



9 Predicted values

[37]:
```
pred_y_df=pd.DataFrame({'Actual value':y_test,'Predicted value':
 ↪y_pred,'Difference':y_test-y_pred})
pred_y_df[0:20]
```

[37]:
```
    Actual value  Predicted value  Difference
0         431.23       431.402451   -0.172451
1         460.01       458.614741    1.395259
2         461.14       462.819674   -1.679674
3         445.90       448.601237   -2.701237
4         451.29       457.879479   -6.589479
5         432.68       429.676856    3.003144
6         477.50       473.017115    4.482885
7         459.68       456.532373    3.147627
8         477.50       474.342524    3.157476
9         444.99       446.364396   -1.374396
10        444.37       441.946411    2.423589
```

```
11        437.04       441.452599   -4.412599
12        442.34       444.746375   -2.406375
13        440.74       440.874598   -0.134598
14        436.55       438.374490   -1.824490
15        460.24       454.370315    5.869685
16        448.66       444.904201    3.755799
17        432.94       437.370808   -4.430808
18        452.82       451.306760    1.513240
19        432.20       427.453009    4.746991
```

```
[ ]:
```