Name: Somit Jain

Reg. No. 20BDS0181

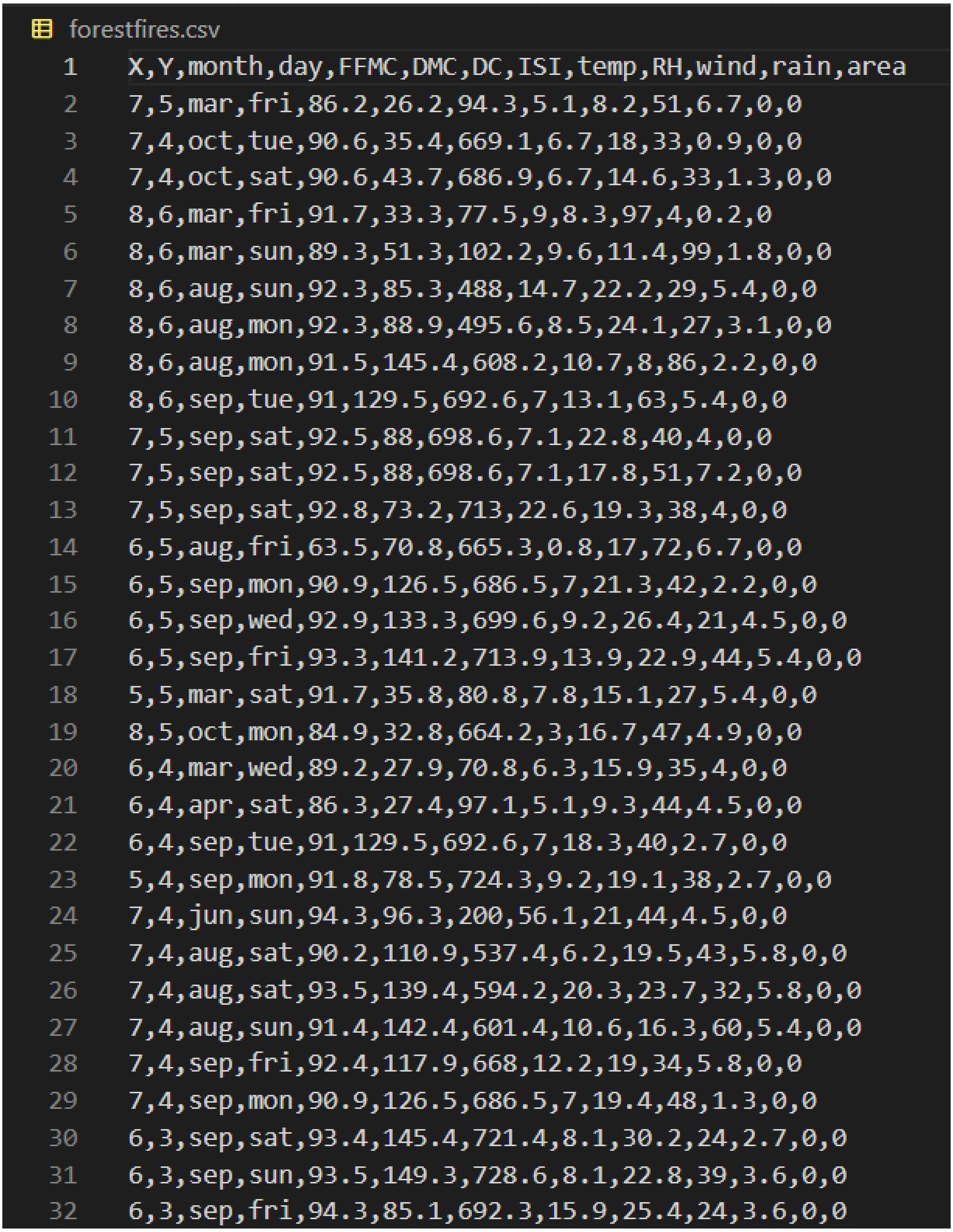
Course: Mathematical Modelling for Data Science(CSE3045 ELA)

Faculty: Dr. Ilanthenral K P S K

Slot: L49+L50

Lab-Assignment 3

DATASET:



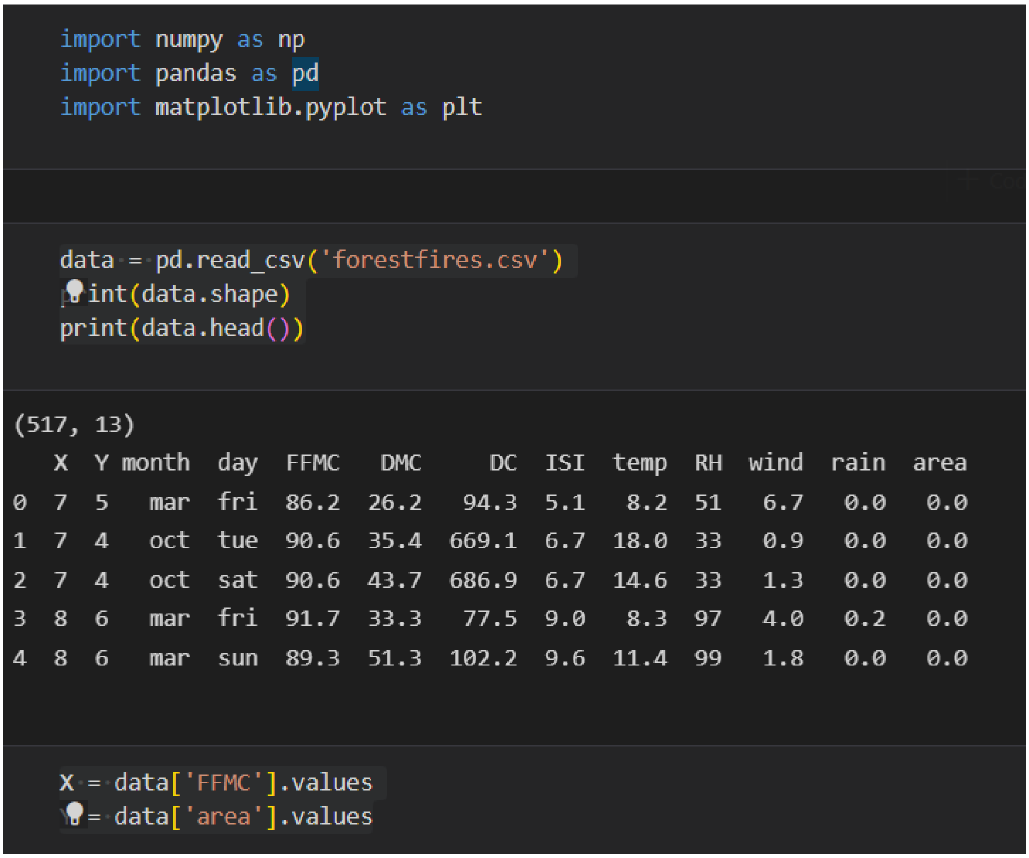
```
X,Y,month,day,FFMC,DMC,DC,ISI,temp,RH,wind,rain,area
7,5,mar,fri,86.2,26.2,94.3,5.1,8.2,51,6.7,0,0
7,4,oct,tue,90.6,35.4,669.1,6.7,18,33,0.9,0,0
7,4,oct,sat,90.6,43.7,686.9,6.7,14.6,33,1.3,0,0
8,6,mar,fri,91.7,33.3,77.5,9,8.3,97,4,0.2,0
8,6,mar,sun,89.3,51.3,102.2,9.6,11.4,99,1.8,0,0
8,6,aug,sun,92.3,85.3,488,14.7,22.2,29,5.4,0,0
8,6,aug,mon,92.3,88.9,495.6,8.5,24.1,27,3.1,0,0
8,6,aug,mon,91.5,145.4,608.2,10.7,8,86,2.2,0,0
8,6,sep,tue,91,129.5,692.6,7,13.1,63,5.4,0,0
7,5,sep,sat,92.5,88,698.6,7.1,22.8,40,4,0,0
7,5,sep,sat,92.5,88,698.6,7.1,17.8,51,7.2,0,0
7,5,sep,sat,92.8,73.2,713,22.6,19.3,38,4,0,0
6,5,aug,fri,63.5,70.8,665.3,0.8,17,72,6.7,0,0
6,5,sep,mon,90.9,126.5,686.5,7,21.3,42,2.2,0,0
6,5,sep,wed,92.9,133.3,699.6,9.2,26.4,21,4.5,0,0
6,5,sep,fri,93.3,141.2,713.9,13.9,22.9,44,5.4,0,0
5,5,mar,sat,91.7,35.8,80.8,7.8,15.1,27,5.4,0,0
8,5,oct,mon,84.9,32.8,664.2,3,16.7,47,4.9,0,0
6,4,mar,wed,89.2,27.9,70.8,6.3,15.9,35,4,0,0
6,4,apr,sat,86.3,27.4,97.1,5.1,9.3,44,4.5,0,0
6,4,sep,tue,91,129.5,692.6,7,18.3,40,2.7,0,0
5,4,sep,mon,91.8,78.5,724.3,9.2,19.1,38,2.7,0,0
7,4,jun,sun,94.3,96.3,200,56.1,21,44,4.5,0,0
7,4,aug,sat,90.2,110.9,537.4,6.2,19.5,43,5.8,0,0
7,4,aug,sat,93.5,139.4,594.2,20.3,23.7,32,5.8,0,0
7,4,aug,sun,91.4,142.4,601.4,10.6,16.3,60,5.4,0,0
7,4,sep,fri,92.4,117.9,668,12.2,19,34,5.8,0,0
7,4,sep,mon,90.9,126.5,686.5,7,19.4,48,1.3,0,0
6,3,sep,sat,93.4,145.4,721.4,8.1,30.2,24,2.7,0,0
6,3,sep,sun,93.5,149.3,728.6,8.1,22.8,39,3.6,0,0
6,3,sep,fri,94.3,85.1,692.3,15.9,25.4,24,3.6,0,0
```

1. Least Square Linear Regression

CODE:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('forestfires.csv')
print(data.shape)
print(data.head())
X = data['FFMC'].values
Y = data['area'].values
from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
A = np.vstack([X,np.ones(len(X))]).T
Y = Y[:, np.newaxis]
alpha = np.dot((np.dot(np.linalg.inv(np.dot(A.T,A)),A.T)),Y)
print(alpha)
plt.figure(figsize = (20,15))
plt.plot(X, Y, 'b.')
plt.plot(X, alpha[0]*X + alpha[1], 'r')
plt.xlabel('X')
plt.ylabel('Y')
plt.xlim(60 , 100)
plt.ylim(-1 , 125)
plt.show()
#using inbuilt method
from sklearn.linear_model import LinearRegression
X = np.c_[data['FFMC']]
Y = np.c_[data['area']]
lin_reg = LinearRegression()
lin_reg.fit(X,Y)
lin_reg.intercept_,lin_reg.coef_
X_new = np.array([18.75,96.119])
B = np.vstack([X_new,np.ones(len(X_new))]).T
Y_predicted = B.dot(alpha)
print(Y_predicted)
plt.figure(figsize = (20,15))
plt.plot(X,Y,"b.")
plt.plot(X_new,Y_predicted,"r-")
plt.xlim(60 , 100)
plt.ylim(-1 , 125)
plt.show()
```

OUTPUT:

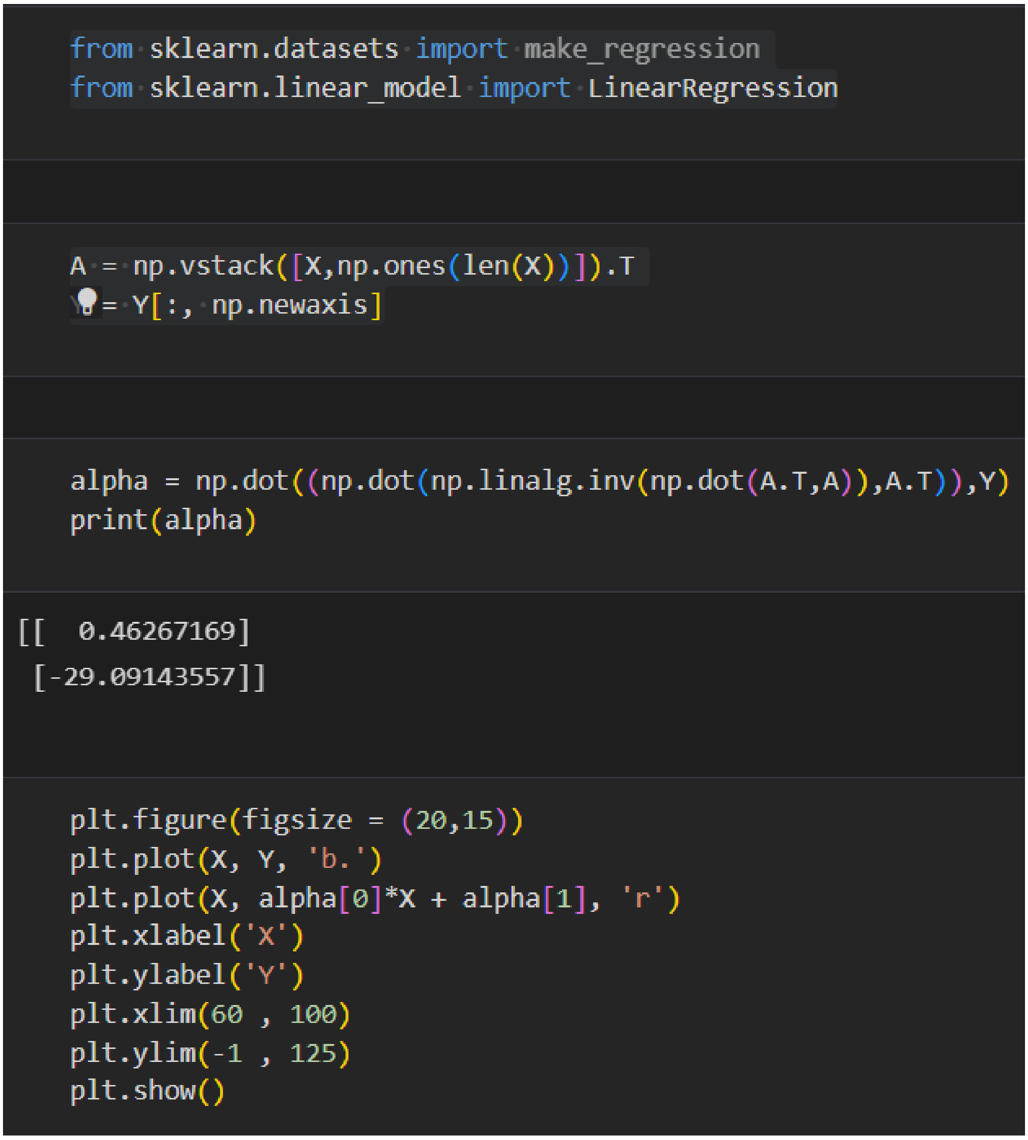


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('forestfires.csv')
print(data.shape)
print(data.head())
```

```
(517, 13)
   X  Y month  day  FFMC   DMC     DC  ISI  temp  RH  wind  rain  area
0  7  5   mar  fri  86.2  26.2   94.3  5.1   8.2  51   6.7   0.0   0.0
1  7  4   oct  tue  90.6  35.4  669.1  6.7  18.0  33   0.9   0.0   0.0
2  7  4   oct  sat  90.6  43.7  686.9  6.7  14.6  33   1.3   0.0   0.0
3  8  6   mar  fri  91.7  33.3   77.5  9.0   8.3  97   4.0   0.2   0.0
4  8  6   mar  sun  89.3  51.3  102.2  9.6  11.4  99   1.8   0.0   0.0
```

```
X = data['FFMC'].values
Y = data['area'].values
```



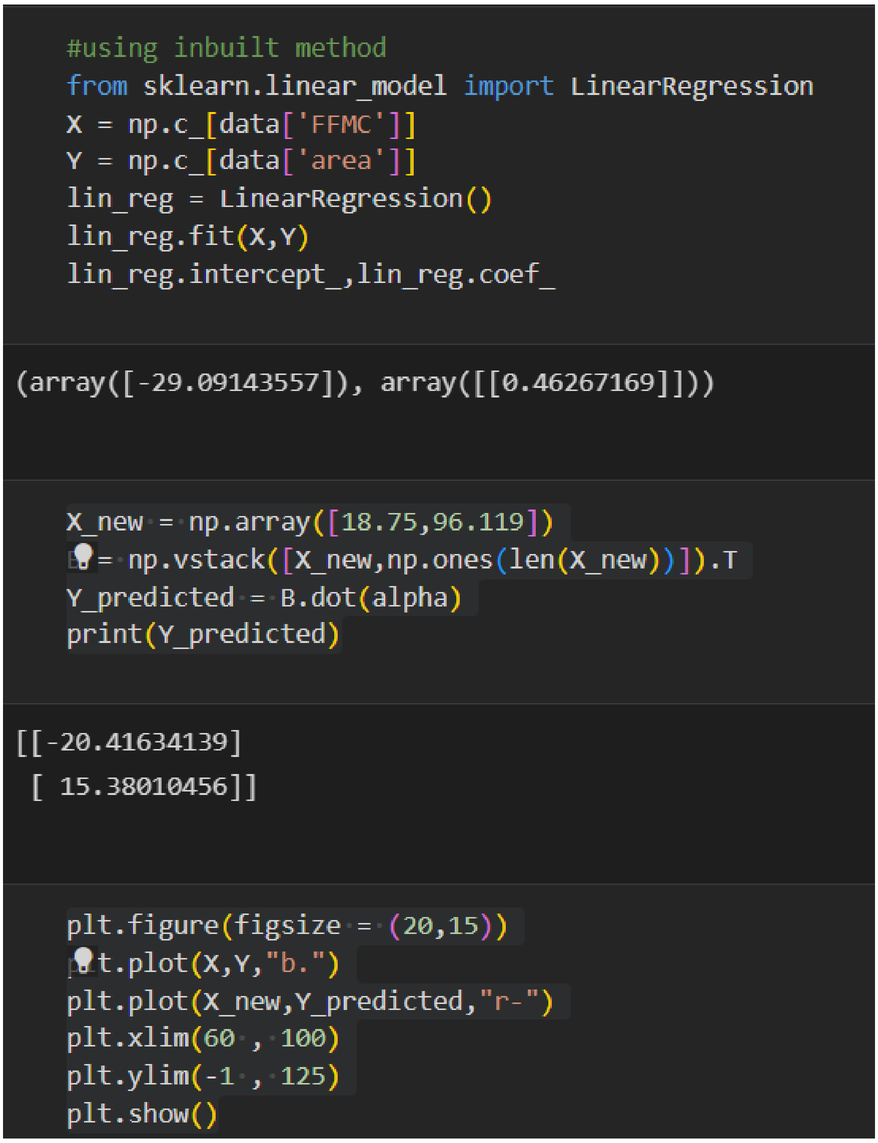
```
from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
```

```
A = np.vstack([X,np.ones(len(X))]).T
Y = Y[:, np.newaxis]
```

```
alpha = np.dot((np.dot(np.linalg.inv(np.dot(A.T,A)),A.T)),Y)
print(alpha)
```

```
[[  0.46267169]
 [-29.09143557]]
```

```
plt.figure(figsize = (20,15))
plt.plot(X, Y, 'b.')
plt.plot(X, alpha[0]*X + alpha[1], 'r')
plt.xlabel('X')
plt.ylabel('Y')
plt.xlim(60 , 100)
plt.ylim(-1 , 125)
plt.show()
```



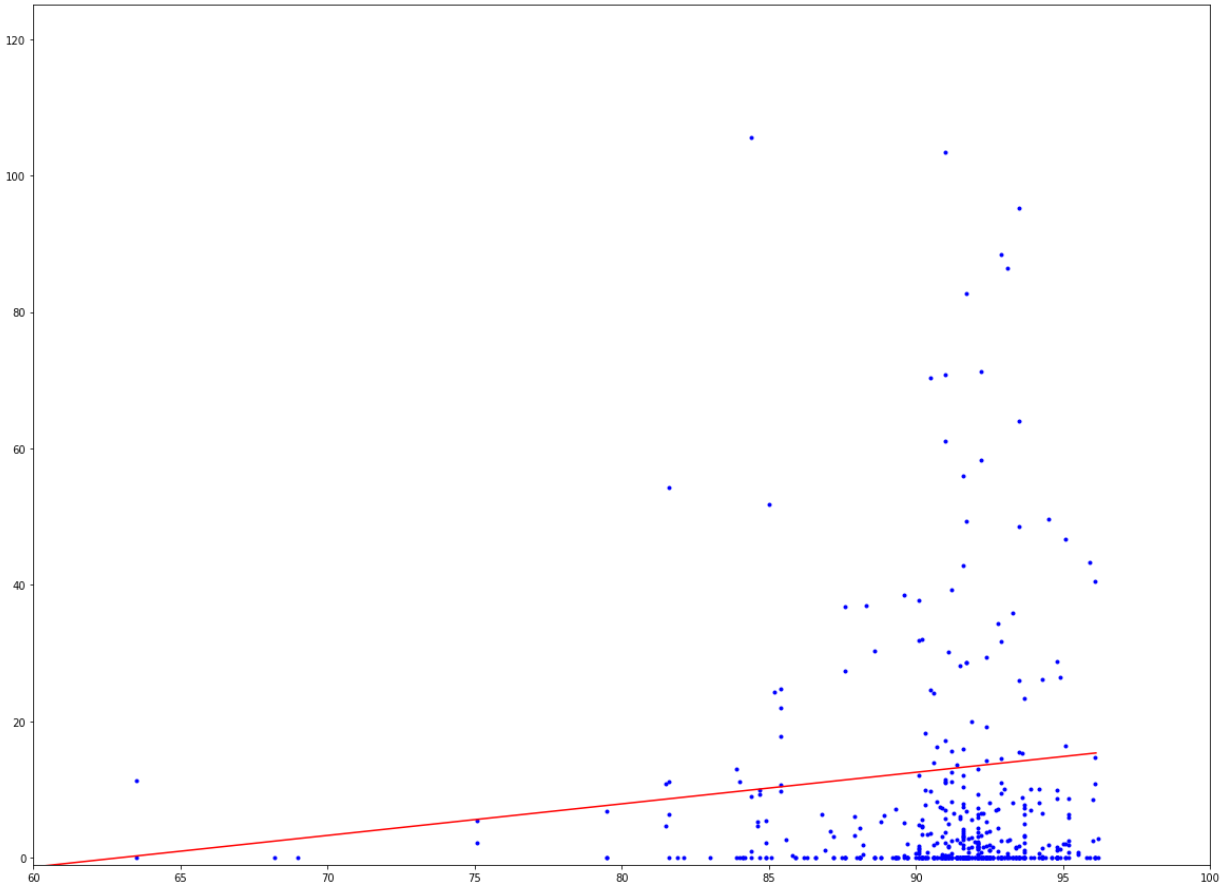
```
#using inbuilt method
from sklearn.linear_model import LinearRegression
X = np.c_[data['FFMC']]
Y = np.c_[data['area']]
lin_reg = LinearRegression()
lin_reg.fit(X,Y)
lin_reg.intercept_,lin_reg.coef_
```

```
(array([-29.09143557]), array([[0.46267169]]))
```

```
X_new = np.array([18.75,96.119])
B = np.vstack([X_new,np.ones(len(X_new))]).T
Y_predicted = B.dot(alpha)
print(Y_predicted)
```

```
[[-20.41634139]
 [ 15.38010456]]
```

```
plt.figure(figsize = (20,15))
plt.plot(X,Y,"b.")
plt.plot(X_new,Y_predicted,"r-")
plt.xlim(60 , 100)
plt.ylim(-1 , 125)
plt.show()
```

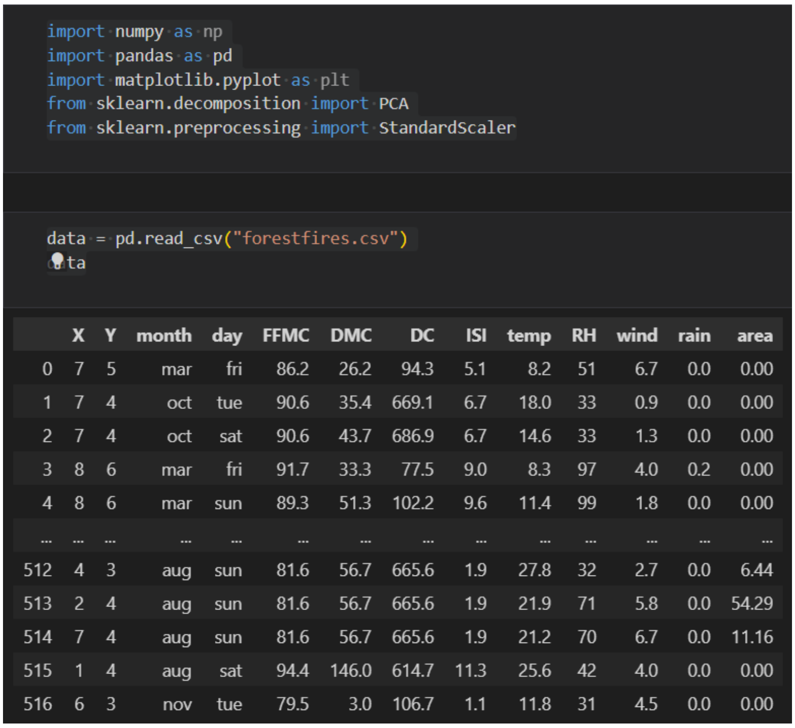


2. Principle Component Analysis

CODE:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
data = pd.read_csv("forestfires.csv")
data
features = data.drop(["X","Y","area"], axis=1)
target = data["area"]
features.head()
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
features['month'] = label_encoder.fit_transform(features['month'])
features['day'] = label_encoder.fit_transform(features['day'])
features.head()
scaler = StandardScaler()
features = scaler.fit_transform(features)
pca = PCA(n_components = 3)
principalComponents = pca.fit_transform(features)
principalDataframe = pd.DataFrame(data = principalComponents, columns = ['PC1', 'PC2','PC3'])
principalComponents
newDataframe = pd.concat([principalDataframe, target],axis = 1)
newDataframe
pca.explained_variance_ratio_
```

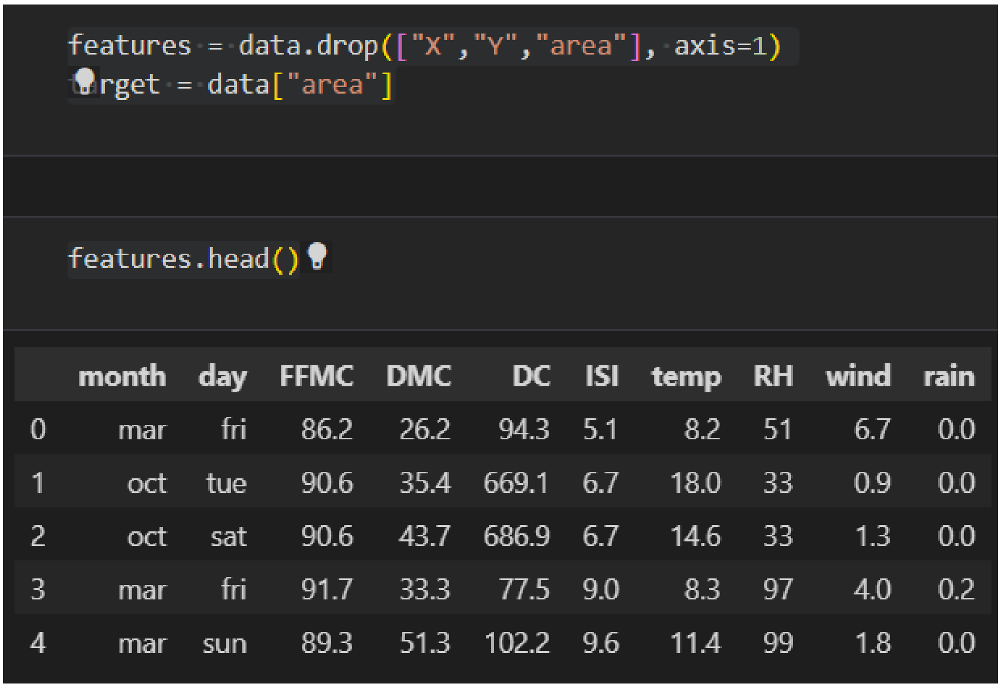
OUTPUT:



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

```
data = pd.read_csv("forestfires.csv")
data
```

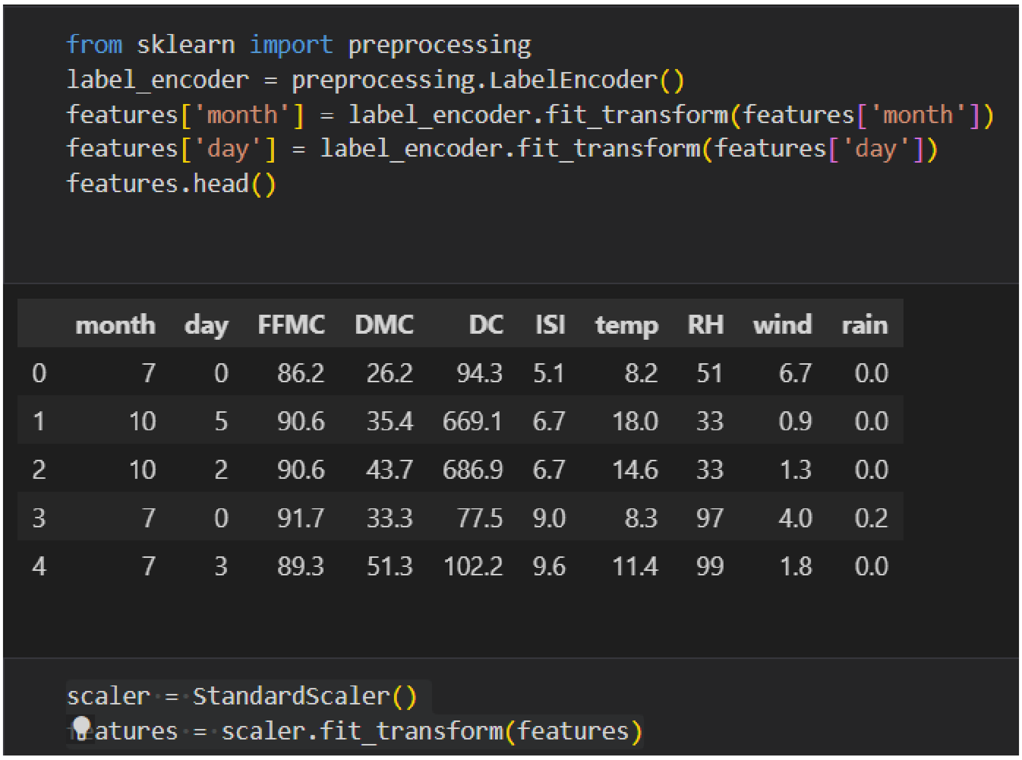
	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...
512	4	3	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
513	2	4	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
514	7	4	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
515	1	4	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
516	6	3	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00



```python
features = data.drop(["X","Y","area"], axis=1)
target = data["area"]
```

```python
features.head()
```

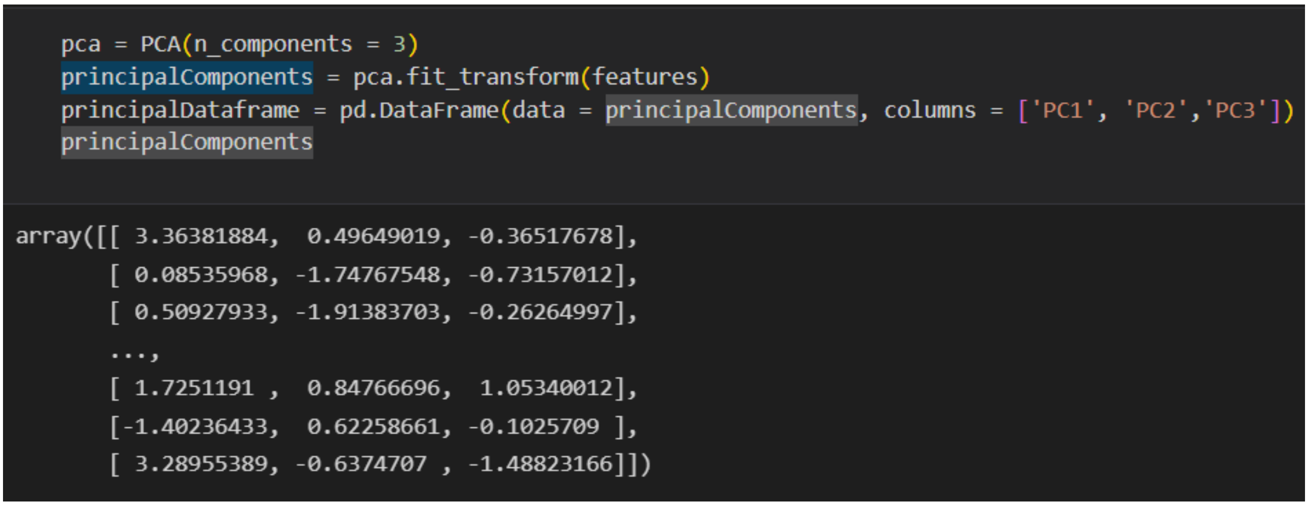
	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain
0	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0
3	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2
4	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0



```python
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
features['month'] = label_encoder.fit_transform(features['month'])
features['day'] = label_encoder.fit_transform(features['day'])
features.head()
```

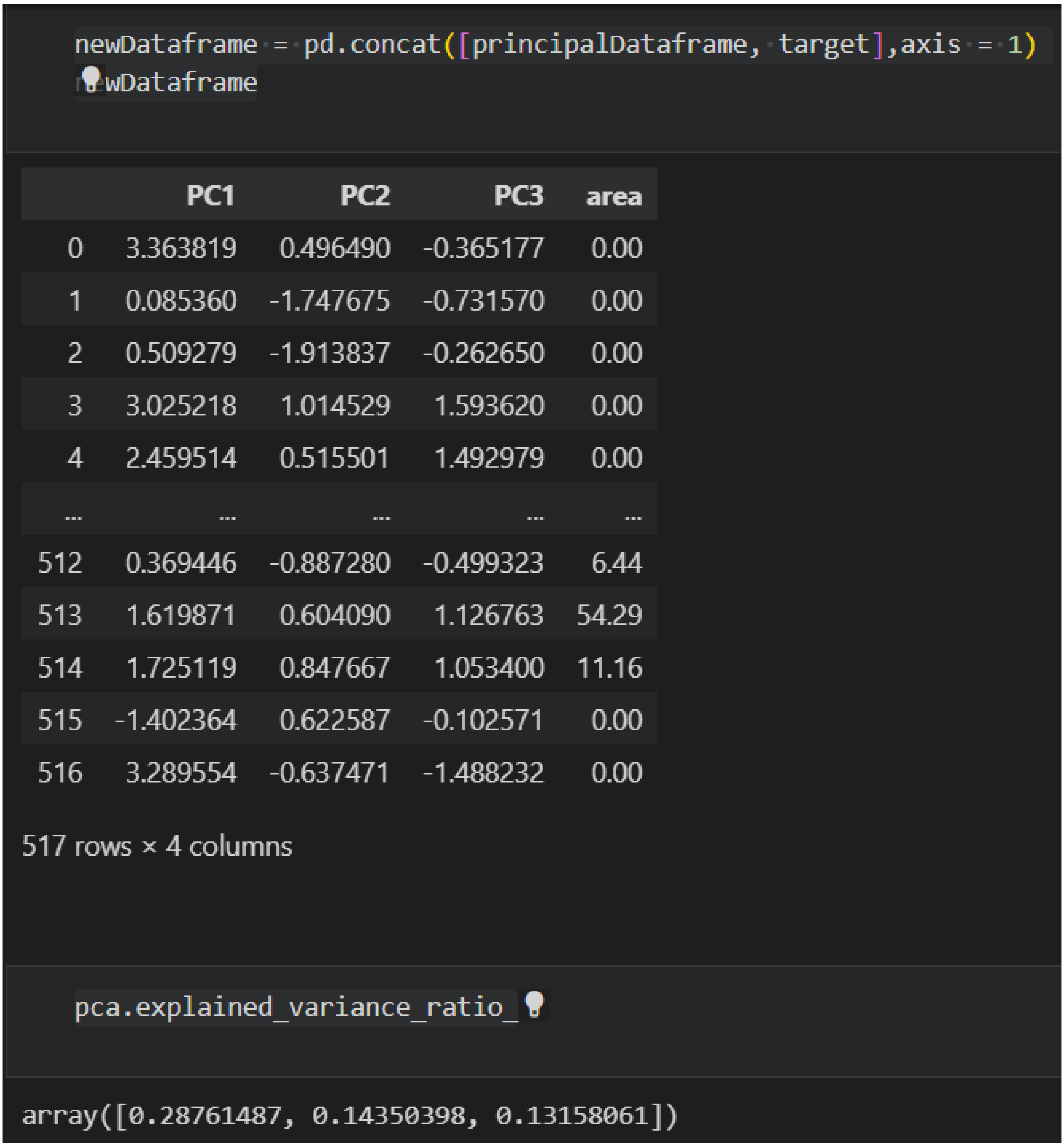
	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain
0	7	0	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0
1	10	5	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0
2	10	2	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0
3	7	0	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2
4	7	3	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0

```python
scaler = StandardScaler()
features = scaler.fit_transform(features)
```



```python
pca = PCA(n_components = 3)
principalComponents = pca.fit_transform(features)
principalDataframe = pd.DataFrame(data = principalComponents, columns = ['PC1', 'PC2','PC3'])
principalComponents
```

```
array([[ 3.36381884,  0.49649019, -0.36517678],
       [ 0.08535968, -1.74767548, -0.73157012],
       [ 0.50927933, -1.91383703, -0.26264997],
       ...,
       [ 1.7251191 ,  0.84766696,  1.05340012],
       [-1.40236433,  0.62258661, -0.1025709 ],
       [ 3.28955389, -0.6374707 , -1.48823166]])
```



```
newDataframe = pd.concat([principalDataframe, target],axis = 1)
newDataframe
```

	PC1	PC2	PC3	area
0	3.363819	0.496490	-0.365177	0.00
1	0.085360	-1.747675	-0.731570	0.00
2	0.509279	-1.913837	-0.262650	0.00
3	3.025218	1.014529	1.593620	0.00
4	2.459514	0.515501	1.492979	0.00
...	...	...	...	...
512	0.369446	-0.887280	-0.499323	6.44
513	1.619871	0.604090	1.126763	54.29
514	1.725119	0.847667	1.053400	11.16
515	-1.402364	0.622587	-0.102571	0.00
516	3.289554	-0.637471	-1.488232	0.00

517 rows × 4 columns

```
pca.explained_variance_ratio_
```

```
array([0.28761487, 0.14350398, 0.13158061])
```