**Name: Somit Jain**

**Reg. No. 20BDS0181**

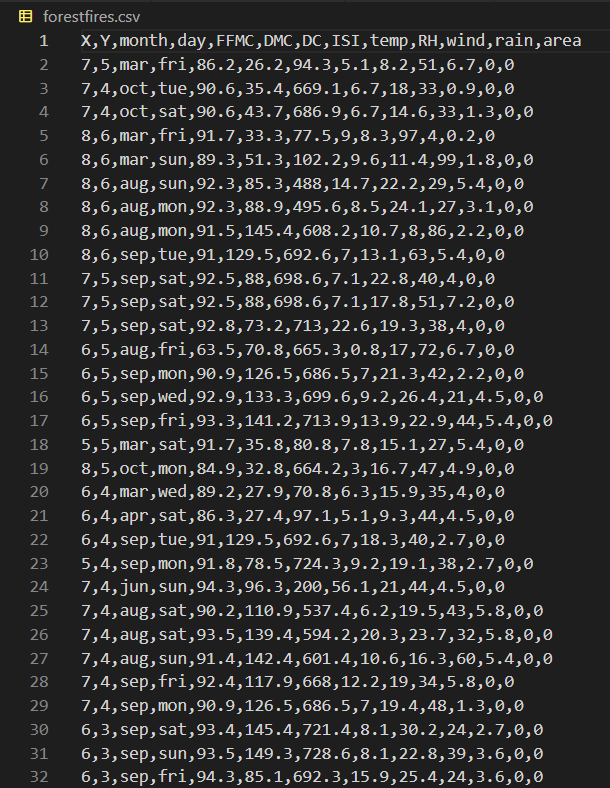
**Course: Mathematical Modelling for Data Science(CSE3045 ELA)**

**Faculty: Dr. Ilanthenral K P S K**

**Slot: L49+L50**

**Lab-Assignment 3**

**DATASET:**



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.")

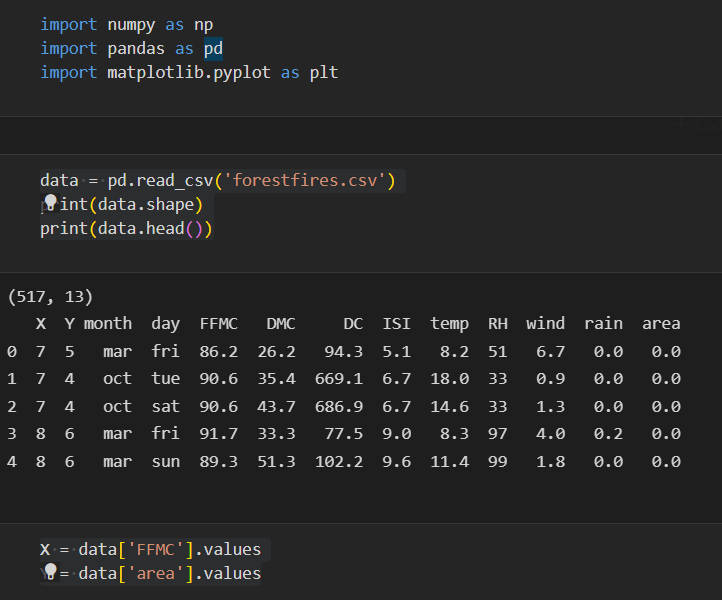
plt.plot(X\_new,Y\_predicted,"r-")

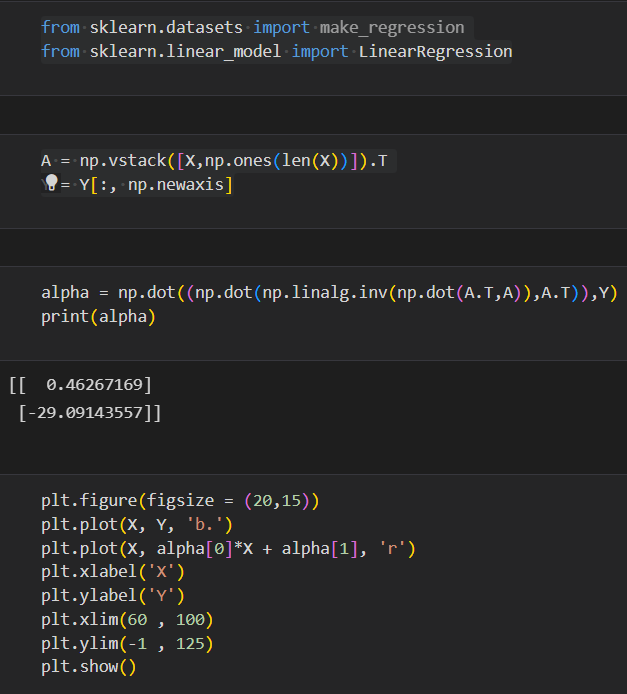
plt.xlim(60 , 100)

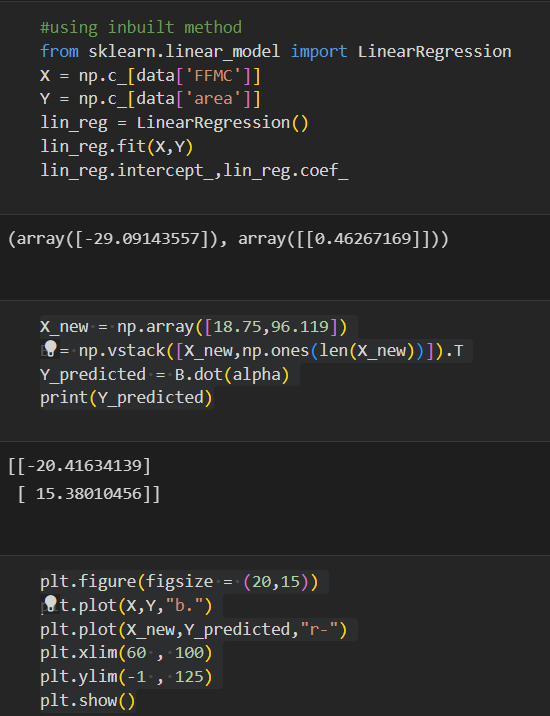
plt.ylim(-1 , 125)

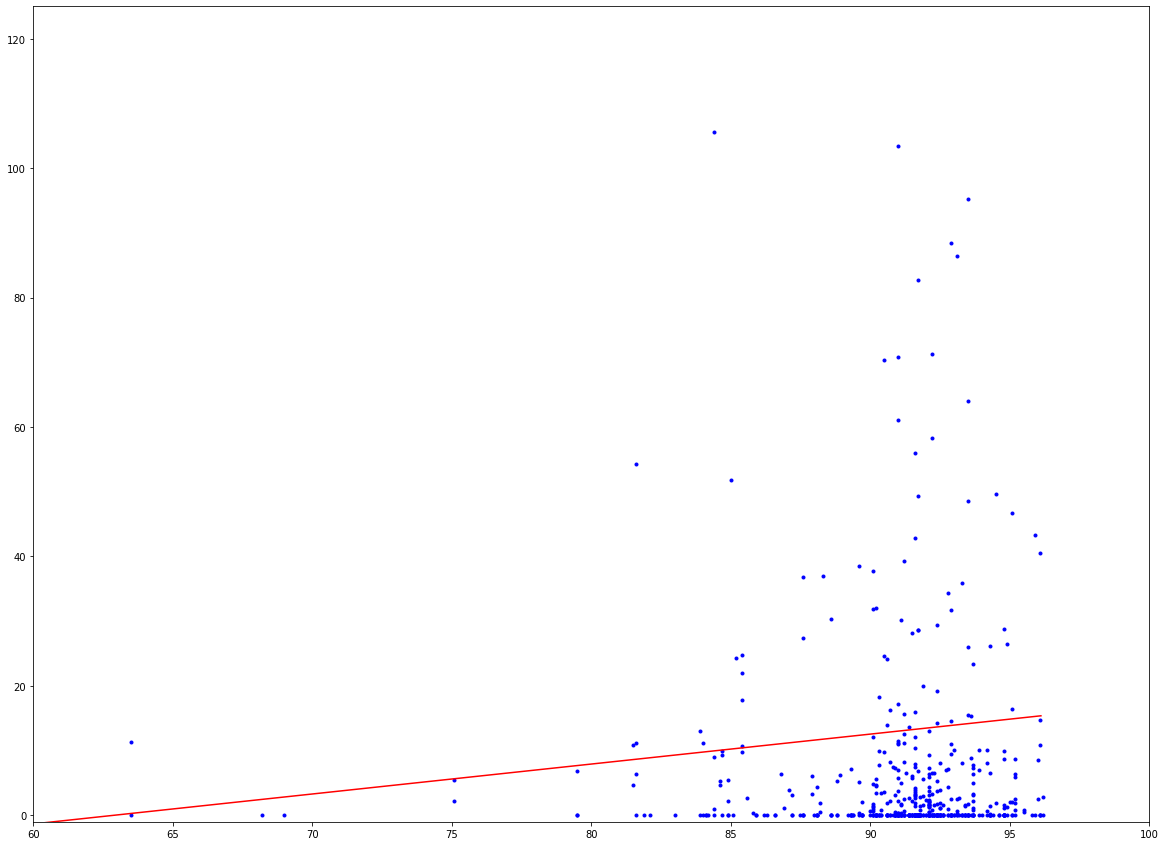
plt.show()

**OUTPUT:**

****

****

****



1. **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:**

