**Write a python program to Implement Linear Regression for predicting salary of the employee given his experience (salary\_data.csv).**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('Downloads/salary\_data.csv')

X = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

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)

from sklearn.linear\_model import LinearRegression

model = LinearRegression()

model.fit(X\_train, y\_train)

predictions = model.predict(X\_test)

predictions

plt.scatter(X\_test, y\_test, color='blue')

plt.plot(X\_train, model.predict(X\_train), color='red')

plt.title('salary of employee vs exp(Test set)')

plt.xlabel('year of exp')

plt.ylabel('salary')

plt.show()

print(model.intercept\_)

print(model.coef\_)

print('X\_test:',X\_test)

print('y\_test:',y\_test)

print('y\_predictions:',predictions)

from sklearn import metrics

print('MAE:', metrics.mean\_absolute\_error(y\_test,predictions))

print('MSE:', metrics.mean\_squared\_error(y\_test,predictions))

print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test,predictions)))

**Write a python program to find out mode and median for the dataset num=[2,3,4,5,7,2]**

1.Mode

from collections import Counter

num = [2,3,4,5,7,2]

no = len(num)

val = Counter(num)

findMode = dict(val)

mode = [i for i, v in findMode.items() if v == max(list(val.values()))]

if len(mode) == no:

findMode = "The group of num not have mode"

else:

findMode = "the mode of number is :"+','.join(map(str,mode))

print(findMode)

2.Median

num = [2,3,4,5,7,2]

no = len(num)

num.sort()

if no % 2 == 0:

median1 = num[no//2]

median2 = num[no//2-1]

median = (median1 + median2)/2

else:

median = num[no//2]

print("The median :" ,str(median))

**Write a python program to find out variance and deviation for the dataset num=[2,3,4,5,7,2]**

import numpy as np

list = [2,4,4,4,5,5,7,9]

print(np.var(list))

print(np.std(list))

print(np.average(list))

**Write a python program to Implement Multiple Regression for predicting CO2 emission of the car given the weight and volume (multiple.csv).**

import pandas

from sklearn import linear\_model

df=pandas.read\_csv("multiple.csv")

X=df[['Weight','Volume']]

y=df['CO2']

regr=linear\_model.LinearRegression()

regr.fit(X,y)

predictedCO2=regr.predict([[2300,1300]])

print(predictedCO2)

1. **Write a python program to fit following data using Polynomial Regression**

**x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]**

**y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]**

import matplotlib.pyplot as plt

import numpy

x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]

y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]

plt.scatter(x,y)

plt.show()

mymodel=numpy.poly1d(numpy.polyfit(x,y,3))

myline=numpy.linspace(1,22,100)

plt.scatter(x,y)

plt.plot(myline,mymodel(myline))

plt.show()

from sklearn.metrics import r2\_score

x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]

y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]

mymodel=numpy.poly1d(numpy.polyfit(x,y,3))

import numpy

import matplotlib.pyplot as plt

x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]

y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]

mymodel=numpy.poly1d(numpy.polyfit(x,y,3))

speed=mymodel(10)

print(speed)

import matplotlib.pyplot as plt

import numpy

x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]

y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]

mymodel=numpy.poly1d(numpy.polyfit(x,y,3))

myline=numpy.linspace(2,95,100)

plt.scatter(x,y)

plt.plot(myline,mymodel(myline))

plt.show()

1. Write a python program to implement prediction of Profit to Production Cost using Decision Tree Regression for below data:

[['Asset Flip', 100, 1000],['Text Based', 500, 3000],['Visual Novel', 1500,5000],['2D Pixel Art', 3500, 8000],['2D Vector Art', 5000, 6500],['Strategy', 6000, 7000],['First Person Shooter', 8000, 15000],['Simulator', 9500, 20000],['Racing', 12000, 21000],['RPG', 14000, 25000],['Sandbox', 15500, 27000],['Open-World', 16500, 30000],['MMOFPS', 25000, 52000],['MMORPG', 30000, 80000]]

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = np.array(

[['Asset Flip', 100, 1000],

['Text Based', 500, 3000],

['Visual Novel', 1500, 5000],

['2D Pixel Art', 3500, 8000],

['2D Vector Art', 5000, 6500],

['Strategy', 6000, 7000],

['First Person Shooter', 8000, 15000],

['Simulator', 9500, 20000],

['Racing', 12000, 21000],

['RPG', 14000, 25000],

['Sandbox', 15500, 27000],

['Open-World', 16500, 30000],

['MMOFPS', 25000, 52000],

['MMORPG', 30000, 80000]

])

print(dataset)

from sklearn.tree import DecisionTreeRegressor

regressor = DecisionTreeRegressor(random\_state = 0)

regressor.fit(X, y)

y\_pred = regressor.predict([[3750]])

print("Predicted price: % d\n"% y\_pred)

X\_grid = np.arange(min(X), max(X), 0.01)

X\_grid = X\_grid.reshape((len(X\_grid), 1))

plt.scatter(X, y, color = 'red')

plt.plot(X\_grid, regressor.predict(X\_grid), color = 'blue')

plt.title('Profit to Production Cost (Decision Tree Regression)')

plt.xlabel('Production Cost')

plt.ylabel('Profit')

plt.show()

import six

import sys

sys.modules['sklearn.externals.six'] = six

from sklearn.externals.six import StringIO

from IPython.display import Image

from sklearn.tree import export\_graphviz

import pydotplus

dot\_data = StringIO()

export\_graphviz(regressor, out\_file =dot\_data,

filled=True, rounded=True,

special\_characters=True,feature\_names =['Production Cost'])

graph =pydotplus.graph\_from\_dot\_data(dot\_data.getvalue())

graph.write\_png('diabetes.png')

Image(graph.create\_png())