***PRACTICAL NO – 3***

**TITLE: POLYNOMIAL LINEAR REGRESSION**

**THEORY:**

In polynomial regression, the dataset when we plot we cannot draw an optimal straight line that covers most of the data points with minimum error like we did it in Linear Regression. Here however we need to find another way of optimisation. Polynomial regression model helps us when our data is non-linear. It creates a curve line trying to fit as many data points as possible. We need to take care that the dataset is not overfitted by increasing the number of degrees. The equation of a polynomial regression is given by

**y = a1 + a1\*x1 + a2\*x12+ a3\*x1+ …**

**CASE 1.**

**PROBLEM STATEMENT**- Prediction of Salary at level 6.5.

**ALGORITHM-**

1.) import required libraries

2.) import the dataset

3.) Divide the dataset into X and Y variables

4.) Fitting the SLR onto the dataset

5.) Fitting PLR onto the dataset

6.) Visualize the dataset on SLR model

7.) Visualize the dataset on PLR model

8.) Predict the value of output at particular instant of input

**CODE: -**

# -\*- coding: utf-8 -\*-

"""

Created on Wed Jul 4 10:11:04 2018

Polynomial Linear Regression

@author: RoKu

"""

"""

the cost function is a polynomial of various degrees of the same feature or multiple

this helps us to get all the featuristic aspects of the feature by fitting not just the

straight line but also the region

"""

"""

Linearity is the aspect of the coefficiants of the equation

"""

#import libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

#import dataset

dataset = pd.read\_csv('position\_salary\_plr.csv')

X = dataset.iloc[:,1:2]

y = dataset.iloc[:,-1]

##fitting the Linear regression model

from sklearn.linear\_model import LinearRegression

linReg = LinearRegression()

linReg.fit(X,y)

from sklearn.preprocessing import PolynomialFeatures

polyReg = PolynomialFeatures(degree=4)

X\_poly = polyReg.fit\_transform(X) ###check the X\_poly it is noe a0,a1 they are x1^0,x1^1,x1^2

polyReg.fit(X\_poly,y)

linreg1 = LinearRegression()

linreg1.fit(X\_poly,y)

##visualizing the both fits

##Linear Regression

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

plt.plot(X,linReg.predict(X),color='blue')

plt.title('Bluff detector(linear regression)')

plt.xlabel = 'Position'

plt.ylabel = 'Salary'

plt.show()

##Polynomial Regression

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

plt.plot(X, linreg1.predict(X\_poly), color='blue')

plt.title('Bluff detector(Polynomial linear regression)')

plt.xlabel = 'Position'

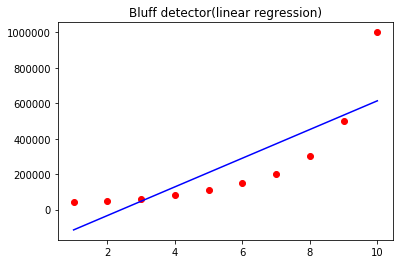
plt.ylabel = 'Salary'

plt.show()

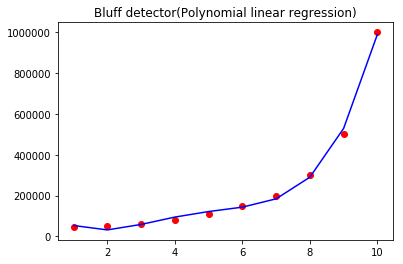
##Predictig the result with Linear regression

print(linreg1.predict(polyReg.fit\_transform(6.5)))

**RESULT:**



**Fiiting a Linear Regressor**



**Fitting a polynomial regressor of degree 4**

**CASE 2.**

**PROBLEM STATEMENT**- Prediction of Petal Length at Sepal length 6.5.

**ALGORITHM-**

1.) import required libraries

2.) import the dataset

3.) Divide the dataset into X and Y variables

4.) Fitting PLR onto the dataset

5.) Visualize the dataset on PLR model

6.) Predict the value of output at particular instant of input

**CODE: -**

"""

Problem Statement:

Find the Item\_Outlet\_Sales for----> item weight = 10.85, item-vis = 0.01589,Item\_MRP = 148

"""

# -\*- coding: utf-8 -\*-

"""

Created on Wed Jul 4 11:55:59 2018

Assignment on PLR

@author: RoKu

"""

#import libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

#import dataset

dataset = pd.read\_csv('Data.csv')

X = dataset.iloc[:,:-1]

y = dataset.iloc[:,1]

from sklearn.linear\_model import LinearRegression

linReg = LinearRegression()

from sklearn.preprocessing import PolynomialFeatures

polyReg = PolynomialFeatures(degree=2)

X\_poly = polyReg.fit\_transform(X)

#Visualizing the polynomial regression results with higher resolution and smmother curve

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

plt.plot(X, linReg.predict(X\_poly), color='blue')

plt.title('Sepal length Predictor')

plt.xlabel = ('Petal Length')

plt.ylabel = ('Species')

plt.grid()

plt.show()

"""

Predicting the sepla length for PetalLength=6.4

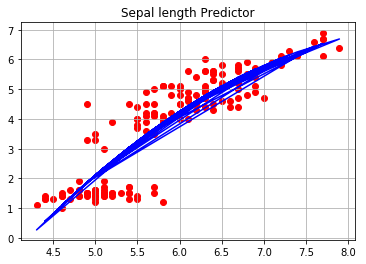
"""

arr = polyReg.fit\_transform(6.4)

print(linReg.predict(arr))

linReg.score(X\_poly,y)

**RESULTS: -**



**Fitting a polynomial of degree = 4**

**DISCUSSIONS:**

The dataset contains temperature readings and pressure readings . We can see that the values are match more accurately as compared to linear regression. At 110 degree celcius of temperature it will give us output pressure as 0.29.At 100 degree pressure is 0.27 which means that we are accurately predicting pressure values.

**CONCLUSIONS:** Successfully acquired the required datasets, compiled the code and got the respective results. The values have been predicted using polynomial linear regression.

|  |  |
| --- | --- |
| **SUBMISSION DATE**-  **15-02-19** | **SIGN OF COURSE INSTRUCTOR**- |
| **ROLL NO OF THE STUDENT-**  **01** | **NAME OF THE STUDENT-**  **Rohit Kulkarni** |