

~\OneDrive\Desktop\ML\_AI\23rd- Poly\1.POLYNOMIAL REGRESSION\POLYNOMIAL.py

```

1 # We will learn slightly advanced ML model called as - POLYNOMIAL REGRESSION MODEL
2 #We did before was simple linear regression and multiple linear regression model
3 #so far we build a linear regressor as linear & multilinear regressor
4
5 #from now we are going to build regressor but that are not linear any more
6 #polynomial regression is not linear regressor, then we build svr, then we build the decision tree regressor & random
7 #forset regression model which are not linear at all
8 #if i use polynomial term in simple linear regression then that is called as polynomial regressor
9 #next svr,dt,random forest based will be based on more complex theory
10
11 # Importing the libraries
12 import numpy as np
13 import matplotlib.pyplot as plt
14 import pandas as pd
15 ****
16
17 # Importing the dataset
18 dataset = pd.read_csv(r"C:\Users\kdata\Desktop\KODI WORK\1. NARESH\2. EVENING BATCH\N_Batch -- 7.30PM\3.
19 MAY\16th,17th\1.POLYNOMIAL REGRESSION\emp_sal.csv")
20 #we are HR team working for big company & we are about hire a new employee, after interview we do see that new employee is
21 #good and fit for the job
22 #we have to give an offer for the potential new employee & now its time to negotiate what's going to be the future salary
23 #of this new employee in the company
24 #so beginning of the negotiation new employee is telling that he has experience and earned 161K annual salary in previous
25 #company, so this employee is asking for more than 161K
26 #however someone in HR team trying to call to the previous employee to check the previous employee information, the
27 #information about the 161k annual salary of future potential of new employee
28 #but unfortunately all the information HR person manage to get from previous employer is the dataset which we are going to
29 #see
30 #dataset is simple table of salary of the 10 different position in the previous company
31 #so the HR member of team runs the simple analysis on excel and observe that there is non -linear relationship b/w the
32 #position level & salary
33 #However the HR person get another very relevant info - new employee has Regional Manager for 2yrs now usually average it
34 #takes 4 yrs to jump from regional manager to partner
35 #so this employee is halfway b/w leve 6 & level 7, so we can say that level is 6.5
36 #so now the HR guy is very much excited he can telling the team to build a regression model to predict the new employee
37 #salary
38 #new employee is telling his annual sal is 161K & lets predict new employee is truth or bluff by using polynomial
39 #regression
40
41 X = dataset.iloc[:, 1:2].values
42 #now we will create X matrix of feature and we will specify the index 1 - LEVEL & however there is something
43 #then i have to mention 2 because in python upper bound of range is excluded
44 y = dataset.iloc[:, 2].values
45 #Dependent variable we will specify the index 2
46
47 #our main goal is to predict if this employee is bluffing by building machine learning model that is polynomial model
48
49 ****

```

```

50
51 # Splitting the dataset into the Training set and Test set
52 #but in this case we dont need to do the training & tesinng , lets see the reason
53 #if you look at the dataset we have only 10 observation & when we have small no. of observation then it doesnot make a
54 sence split the data into trainin & testing
55 #so we dont have enough information to train the model
56 #onter point is we want to make very accurate prediction becuasewe are trying to predict the salary of new employee, if we
57 not build the accurate the prediction
58 #in order to make accurate prediction we need to have maximum observation so that model will get perfectly get the
59 corelation b/w the dataset
60 #only for this time we will allow ourself to take whole dataset to train the ML model
61
62 ****
63 #feature scaling is also not required on this case becuase we have to add polynomial function right
64 # also we will use same linear regression library for this also
65 # so no need to feature scaling and no need to training and testing also
66
67 ****
68 #In this section we build the linear regression model & also we gonna build the polynomial regression model to fit the
69 dataset
70 #why we creating the 2 models becasae if we go backe to the equation & also compare the both results
71
72 # Fitting Linear Regression to the dataset
73
74 from sklearn.linear_model import LinearRegression
75 lin_reg = LinearRegression()
76 #To compare both linear model created lin_reg & for polynomial we will mention as lin_reg2
77 #so create a object of lin_reg and called the class LinearRegression
78 lin_reg.fit(X, y)
79 #fit the lin_reg object to X & y. now our simple linear regression is ready
80
81 ****
82 #Now lets build the polynomial regression model
83 #to create this model we will import a PolynomialFeature
84
85 # Fitting Polynomial Regression to the dataset
86 from sklearn.preprocessing import PolynomialFeatures
87 #preprocessing is library we used for feature scaling in data preprocessing
88 #from preprocessing library we called the class as PolynomialFeatures class
89 poly_reg = PolynomialFeatures() # we mentoine 2 degree
90 #create an object called poly_reg & we will assigned the degree
91 #what exactly poly_reg will do the transform the feature of X & y to new matrix called X_poly & y_poly
92 #polyfeature will transfer from X1 to X1square, X2 to X2 square
93 #it will transform untill you mention the power
94 #in one independent variable X till 2,3,4 or 5 columns will be crreated based on the power
95 #to the first step i will mention the default degree is 2
96 X_poly = poly_reg.fit_transform(X)
97 #finaly we will create new X_poy object or X_poly matrix to hold newly created columns by polyfeatures
98 #poly_reg is the object to fit and transform from X to X_poly matrix
99 #we are transform from X independent to poly by adding degree 2
100
101 ****
102 #Lets see the variable explore what happend hear first upol click on X you get the orighinal level & in X_poly matrix we
103 have 3 columns
104 #in the X_poly where is our X which is our second column & if you see 2nd then you get the degree of square
105 #remember in multiple linear regression we have to add manaully of column 1 called constant
106 #when you called Poly_reg_transform automatically or bydefault 1 will be created
107
108 ****

```

```

109
110 lin_reg_2 = LinearRegression()
111 #we crate an 2nd object for same LinearRegression
112 lin_reg_2.fit(X_poly, y)
113 #now we need to fit lineare regression object to X_poly & y
114 #now our polynomial model is created & we will ready to review the truth or bluff part
115 #now we build 2 regressor - linear regressor and polynomial regressor
116
117 ****
118
119 #now we gonna visualize all the result & we will start all the observation point on x-asix and we will 10 related salary
120 #on y-axis
121 #first we will draw the visualtin of linear regression model and then we will build the visualize of polynomial regression
122 #model
123 #we will compare the true value of true observaiton made by the model
124
125 # Visualising the Linear Regression results
126
127 #lets starts the plotting by true observation
128 plt.scatter(X, y, color = 'red')
129 #we are going to plot for actual value of X & y
130 plt.plot(X, lin_reg.predict(X), color = 'blue')
131 #now plot for the prediction line where x coordinate are predictin points & for y-cordinates predicted value which is
132 lin_reg.predict (x)
133 plt.title('Truth or Bluff (Linear Regression)')
134 plt.xlabel('Position level')
135 plt.ylabel('Salary')
136 plt.show()
137
138 #now we visualize the linar regression graph,lets check the difference b/w actual point & predicted point
139 #prediction of linear regression model is straight line, red - real observation point & blue is predicted observation
140 #point
141 #this is not a right predicton, as i said linear model if you applied on non-linear dataset then you wont get the accurate
142 #predictino
143 #only one point close to the prediction & one point is very far from the prediction line
144 #lets take one example is if you take top point actual sal is 100million but predicte is 60000 which had some error rate
145 #thats why we need the batter model & this is linear model prediction
146 #next step is we will build the non-linear model
147
148 ****
149
150 # Visualising the Polynomial Regression results
151 plt.scatter(X, y, color = 'red')
152 plt.plot(X, lin_reg_2.predict(poly_reg.fit_transform(X)), color = 'blue')
153 #in y-coordinate we have to replace with lin_reg2 which we create for poly regression model
154 #X_poly is not defined cuz we already defined in above plot, so insted of X_poly we will define complete fit_trasnform
155 #code
156 plt.title('Truth or Bluff (Polynomial Regression)')
157 plt.xlabel('Position level')
158 plt.ylabel('Salary')
159 plt.show()
160
161 #After execute first thing we dont have straing line hear & we got curve hear & we can say that polynomial regression is
162 #not a linear regression model
163 #you can easily say that distinguish b/w linear model & non-linear model
164 #linear model is straigt line but non-linear is not a straight line
165 # lets take the higst point which is ceo whose actual sal is 100k but found predicted as 90k, but better then last time
166 #still we can say bit improved not accurate model which model thots the reason we have to increase the degree from 2 to 3
167 #if you check what is the predicted sal of 6.5 yr of exper whose sal was 161k but as per model predict that almost equal
168 #so we can say that employee is honest
169 # blue curve is much better complare to last then the linear regression model
170 #now we can do much better by adding degree from 2 to 3
171
172 #only you have to do is adding instead of 3 for better improvement
173 #our loop is approaching is much better , and you can say that poly model is quite improving

```

```
166 #now if you observe that ceo actual salar is almost try to equal to predicted & based on this situation you can say that  
167 can we imporve the model bit more  
168  
169 #next step you have to do is adding one more degree to 4 & now you get that blue curve is almost fit to the actual point  
170 and you can that employee is genuine  
171  
172 # Final step to do and that is weeather the new employee is truth or bluff by predicting by polynomial  
173 #new empoyy previous sal is 161k , we will compare this value to our prediction & we will see if the employee was bluffing  
174 *****  
175  
176 # Predicting a new result with Linear Regression  
177 lin_reg.predict([[6.5]])  
178 #means that show me the predicted salary coresponding to 6.5 & lets see what our linear regression model is predict  
179 #if you go back to linear plot 6.5 level we got as 331k somewher around & this is linear regression result and this is not  
180 the definitely not the best one for prediction  
181  
182 # Predicting a new result with Polynomial Regression  
183 lin_reg_2.predict(poly_reg.fit_transform([[6.5]]))  
184 #This code show me that predicted salary of 6.5 level using poly reg model  
185 #That means employee is True and we solved this by using polyregression model  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198
```