

~\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
  forset regression model which are not linear at all
7 #if i use polynomial term in simple linear regression then that is called as polynomial regressor
8 #next svr,dt,random forest based will be based on more complex theory
9
10 # Importing the libraries
11 import numpy as np
12 import matplotlib.pyplot as plt
13 import pandas as pd
14
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.
  MAY\16th,17th\1.POLYNOMIAL REGRESSION\emp_sal.csv")
19 #we are HR team working for big company & we are about hire a new employee, after intervei w we do see that new employee is
  good and fit for the job
20 #we have to give an offer for the potential new employee & now its time to negotiate whats going to be the future salary
  of this new employee in the company
21 #so begining of the negotiation new employee is telling that he has experience and earned 161K annual salary in previous
  company, so this employee is asking for more then 161K
22 #however someone in HR team trying to call to the previous employee to check the previous employee information, the
  information about the 161kannual salary of future potential of new employee
23 #but unfortunately all the information HR person manage to get from previous employer is the dataset which we are going to
  see
24 #dataset is simple table of salary of the 10 different position in the previous company
25 #so the HR member of team runs the simple analysis on excel adn observe that there is non -linear relationshipb/w the
  position level & salary
26 #However the HR person get another very relavant info - new employee has Regional Manager for 2yrs now usually average it
  takes 4 yrs to jump from regional manager to partner
27 #so this employee is halfway b/w leve 6 & level 7, so we can say that level is 6.5
28 #so now the HR guy is very much excited he can telling the team to build a regression model to predict the new employee
  salary
29 #new employee is telling his annual sal is 161K & lets predict new employee is truth or bluff by using polynomial
  regression
30
31 #now lets see what index we need to check, so when we look at the table hear we want to predict the salary based on
  different level and then we predict salary of 6.5 level
32 #we dont need the categorical column becuase that column is equivalent to level only
33 #we encoded the positio level associated with each position level
34 #so we need 2 columns to build the ML model & machine will learn the corelation b/w position and salary to predict if the
  employee is bluffing about salary
35 #so we dont need the categorical data & we consider only 2 columns
36 #so our X-matrix is only one column is level and y-matrix is salary
37 #this dataset contains 3 columns - position , level & salary
38 #if you take indexing from column wise then postion would be 0 index - position,1st index is lvele & 2nd index one is
  salary indexing
39
40
41 X = dataset.iloc[:, 1:2].values
42 #now we will create X matrix of feature and we will spcify 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
  sence split the data into trainin & testing
54 #so we dont have enough information to train the model
55 #onter point is we want to make very accurate prediction becuasewe are trying to predict the salary of new employee, if we
  not build the accurate the prediction
56 #in order to make accurate prediction we need to have maximum observation so that model will get perfectly get the
  corelation b/w the dataset
57 #only for this time we will allow ourself to take whole dataset to train the ML model
58
59 *****
60
61 #feature scaling is also not required on this case becuase we have to add polynomial function right
62 # also we will use same linear regression library for this also
63 # so no need to feature scaling and no need to training and testing also
64
65 *****
66 #In this section we build the linear regression model & also we gonna build the polynomial regression model to fit the
  dataset
67 #why we creating the 2 models becusae if we go backe to the equation & also compare the both results
68
69 # Fitting Linear Regression to the dataset
70
71 from sklearn.linear_model import LinearRegression
72 lin_reg = LinearRegression()
73 #To compare both linear model created lin_reg & for polynomial we will mention as lin_reg2
74 #so create a object of lin_reg and called the class LinearRegression
75 lin_reg.fit(X, y)
76 #fit the lin_reg object to X & y. now our simple linear regression is ready
77
78 *****
79
80 #Now lets build the polynomial regression model
81 #to create this model we will import a PolynomialFeature
82
83 # Fitting Polynomial Regression to the dataset
84 from sklearn.preprocessing import PolynomialFeatures
85 #preprocessing is library we used for feature scaling in data preprocessing
86 #from preprocessing library we called the class as PolynomialFeatures class
87 poly_reg = PolynomialFeatures() # we mentoine 2 degree
88 #create an object called poly_reg & we will assigned the degree
89 #what exactly poly_reg will do the transform the feature of X & y to new matrix called X_poly & y_poly
90 #polyfeature will transfer from X1 to X1square, X2 to X2 square
91 #it will transferm untill you mention the power
92 #in one independent variable X till 2,3,4 or 5 columns will be crreated based on the power
93 #to the first step i will mention the default degree is 2
94 X_poly = poly_reg.fit_transform(X)
95 #finaly we will create new X_poy object or X_poly matrix to hold newly created columns by polyfeatures
96 #poly_reg is the object to fit and transform from X to X_poly matrix
97 #we are transform from X independent to poly by adding degree 2
98
99 #Lets see the variable explore what happend hear first upol click on X you get the original level & in X_poly matrix we
  have 3 columns
100 #in the X_poly where is our X which is our second column & if you see 2nd then you get the degree of square
101 #remember in multiple linear regression we have to add manually of column 1 called constant
102 #when you called Poly_reg_transform automatically or bydefault 1 will be created
103
104 poly_reg.fit(X_poly, y)
105 #now we have to fit the poly_reg fit instad of x we have to fit with X_poly, y
106 #in linear we have done lin_reg.fit (x,y)
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
    on y-axis
120 #first we will draw the visualtin of linear regression model and then we will build the visualize of polynomial regression
    model
121 #we will compare the true value of true observaiton made by the model
122
123 # Visualising the Linear Regression results
124
125 #lets starts the plotting by true observation
126 plt.scatter(X, y, color = 'red')
127 #we are going to plot for actual value of X & y
128 plt.plot(X, lin_reg.predict(X), color = 'blue')
129 #now plot for the prediction line where x coordinate are predictin points & for y-coordinates predicted value which is
    lin_reg.predict (x)
130 plt.title('Truth or Bluff (Linear Regression)')
131 plt.xlabel('Position level')
132 plt.ylabel('Salary')
133 plt.show()
134
135 #now we visualize the linar regression graph,lets check the difference b/w actual point & predicted point
136 #prediction of linear regression model is straight line, red - real observation point & blue is predicted observation
    point
137 #this is not a right predicton, as i said linear model if you applied on non-linear dataset then you wont get the accurate
    predictino
138 #only one point close to the prediction & one point is very far from the prediction line
139 #lets take one example is if you take top point actual sal is 100million but predicte is 60000 which had some error rate
140 #thats why we need the batter model & this is linear model prediction
141 #next step is we will build the non-linear model
142
143 *****
144
145 # Visualising the Polynomial Regression results
146 plt.scatter(X, y, color = 'red')
147 plt.plot(X, lin_reg_2.predict(poly_reg.fit_transform(X)), color = 'blue')
148 #in y-coordinate we have to replace with lin_reg2 which we create for poly regression model
149 #X_poly is not defined cuz we already defined in above plot, so insted of X_poly we will define complete fit_trasnform
    code
150 plt.title('Truth or Bluff (Polynomial Regression)')
151 plt.xlabel('Position level')
152 plt.ylabel('Salary')
153 plt.show()
154
155 #After execute first thing we dont have straing line hear & we got curve hear & we can say that polynomial regression is
    not a linear regression model
156 #you can easily say that distinguish b/w linear model & non-linear model
157 #linear model is straight line but non-linear is not a straight line
158 # lets take the highst point which is ceo whose actual sal is 100k but found predicted as 90k, but better then last time
159 #still we can say bit improved not accurate model which model thats the reason we have to increase the degree from 2 to 3
160 #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
    so we can say that employee is honest
161 # blue curve is much better complare to last then the linear regression model
162 #now we can do much better by adding degree from 2 to 3
163
164 #only you have to do is adding instead of 3 for better improvement
165 #our loop is approching 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
    can we imporve the model bit more
167 #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
    and you can that employee is genuine
168
169 # Final step to do and that is weeather the new employee is truth or bluff by predicting by polynomial
170 #new empoyy previous sal is 161k , we will compare this value to our prediction & we will see if the employee was bluffing
171
172 #*****
173
174 # Predicting a new result with Linear Regression
175 lin_reg.predict([[6.5]])
176 #means that show me the predicted salary coresponding to 6.5 & lets see what our linear regression model is predict
177 #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
    the definitely not the best one for prediction
178
179 # now we will see the salary of 6.5 level what is the prediction salary part
180
181 # Predicting a new result with Polynomial Regression
182 lin_reg_2.predict(poly_reg.fit_transform([[6.5]]))
183 #This code show me that predicted salary of 6.5 level using poly reg model
184 #That means employee is True and we solved this by using polyregression model
185
186
187
188
189
190
191
192
193
194
195
196
197
198
```