Polynomial Regression

Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial

Polynomial Regression algorithm is also called Polynomial Linear Regression because it does not depend on the variables, instead, it depends on the coefficients, which are arranged in a linear fashion

Formula - y= b0+b1x1+ b2x12+ b2x13+...... bnx1n

Why Polynomial Regression

If we apply a linear model on a linear dataset, then it provides us a good result as we have done in Linear Regression, but if we apply the same model without any modification on a non-linear dataset, then it will produce a drastic output.

code -

import pandas as pd

data\_set= pd.read\_csv('Position\_Salaries.csv')

x= data\_set.iloc[:, 1:2].values

y= data\_set.iloc[:, 2].values

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import PolynomialFeatures

poly\_regs= PolynomialFeatures(degree= 2)

x\_poly= poly\_regs.fit\_transform(x)

lin\_reg =LinearRegression()

lin\_reg.fit(x\_poly, y)

KNN Algorithm

K-Nearest Neighbour is Supervised Learning technique.

K-NN algorithm assumes the similarity between the new case and available cases and put the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

Working-

Step-1: Select the number K of the neighbors # Large values for K are good

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

Code-

import pandas as pd

#importing datasets

data\_set= pd.read\_csv('data.csv')

x= data\_set.iloc[:, [1,3]].values

y= data\_set.iloc[:, 4].values

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test= train\_test\_split(x, y, test\_size= 0.25, random\_state=0)

from sklearn.preprocessing import StandardScaler

st\_x= StandardScaler()

x\_train= st\_x.fit\_transform(x\_train)

x\_test= st\_x.transform(x\_test)

from sklearn.neighbors import KNeighborsClassifier # implement KNN

classifier= KNeighborsClassifier(n\_neighbors=5, p=2 ) # n\_neighbors: To define the required neighbors of the algorithm.

classifier.fit(x\_train, y\_train)

y\_pred= classifier.predict(x\_test)