Lab programs 15/3/2022

9) Non parametric Locally weighted Regression Algorithm (15 lines)

Line NO →

→ Python library to plot graphs

→ Alias

1) import matplotlib.pyplot as plt

2) from scipy.interpolate import interp1d

method to create function based on fixed data point

it takes x and y returns callable function by using x and returns y

→ Alias

3) import statsmodels.api as sm

provides classes and models to estimate different statastical model

→ temporary variable to store current position

→ total noof y

4) x = [i/5.0 for i in range(30)]

5) y= [1, 2, 1, 2, 1, 1, 3, 4, 5, 4, 5, 6, 5, 6, 7, 8, 9, 10, 11, 11, 12, 11, 11, 10, 12, 11, 11, 10, 9, 13] → return value

6) lowess = sm.nonparametric.lowess(y, x)

locally weighted scatterplot smoothening.

alias

strategy to fit smooth curve to datapoint

zip(*) takes iterable and return iterator

→ smooten x

7) lowess_x = list(zip(*lowess))[0]

8) lowess_y = list(zip(*lowess))[1]

→ smoothen y

→ store object

9) f = interp1d(lowess_x, lowess_y, bounds_error= False)

run scipy interpolation.

skip error values

10) xnew = [i/10.0 for i in range(100)]

new width after Smoothening

11) y new = f[x new]

temp variable to store scipy interpolate

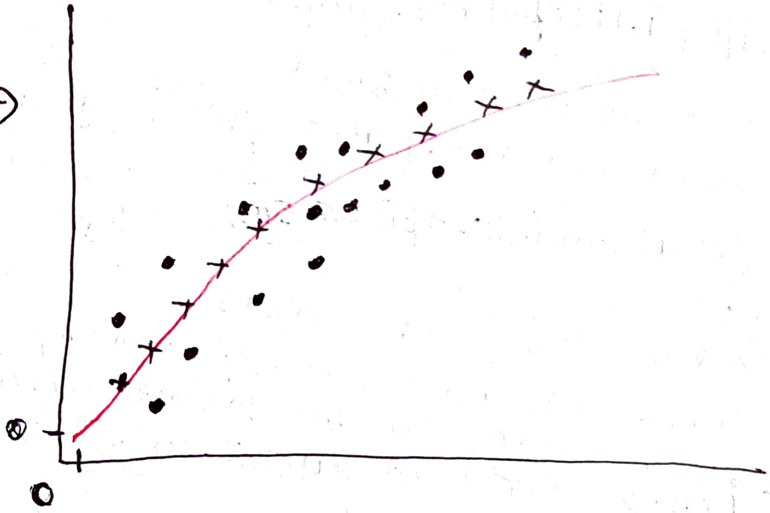
12) plt.plot (x, y, 'o') → All x marks

13) plt.plot (lowess_x, lowess_y, '+') → red line

14) plt.plot (xnew, ynew, '-') → dots.

15) plt.show()

output →



⑥ Program to implement Naive Bayesian Classifier for a sample training

1) import pandas as pd

here output is dataset

Panda is used when we deal with machine learning task

2) PlayTennis = pd.read-csv("4.csv")

variable ML. read function located in same folder

3) print("Given Dataset is: \n", PlayTennis, "\n")

print print given dataset. new line

vast library

4) from Sklearn.preprocessing import LabelEncoder()

find

5) Le = LabelEncoder() → converts non numeric value to numeric 0 to n-1

Store into Le

convert non numeric to numeric

variable individual attribute variable

6) PlayTennis['outlook'] = Le.fit_transform(PlayTennis['outloo

7) PlayTennis['temp'] = Le.fit_transform (PlayTennis['temp'])

8) PlayTennis ['humidity'] = Le.fit-transform(PlayTennis['humidity'])

9) PlayTennis ['wind'] = Le.fit-transform (PlayTennis ['wind'])

10) PlayTennis ['play'] = Le.fit-transform(PlayTemis ['play'])

mistake i used pd.

print.

11) print("The Encoded dataset is \n", PlayTennis)

after encode print.

drop 1 axis play column is not required

12) x = PlayTennis.drop(['play'], axis = 1)

13) y = playTennis ['play']

i missed here.

drop axis from both

Vast library

14) from Sklearn.model_ import train_test_split →forgot
Selection

divide dataset into train & test

15) from sklearn.naive_bayes import GaussianNB

Special Algorithm

16) from sklearn.metrics import accuracy_score

accurate value

Store split value.

17) x_train, x_test, y_train, y_test =
train_test_split (x, y, test_size =0.20)

function to split

it should be between 0.0 to 1.0

print

18) print("\n x_train:\n", x_train)] train data
19) print("\n y_train: \n", y_train)

20) print("\n x_test \n", x_test)] test data
21) print("\n y_test \n", y_test)

miss maadidre 0.6 accuracy

22) classifier = GaussianNB()

23) classifier.fit (x_train, y_train) → forgot

convert train data into Gaussian NB

24) →forgot
accuracy = accuracy_score(classifier.predict(x_test), y_test)

print

25) print("\n Accuracy is :", accuracy)

Output →

1) Given dataset
2) Encoded dataset
3) train and test data
4) Accuracy.

ram