**[y ni value fix than Classification , y ni value continue than regression]**

**First identify x y Than dependent or If y is contain continue (emp salary fix no hoy value)Regression**

**Type of flower fix value (Classification)**

**REGRESSION LINEAR**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn import linear\_model

#load the data

data=pd.read\_csv("Salary.csv")

print(data)

#split the dataset

x=np.array(data.YearsExperience).reshape(-1,1)

y=np.array(data.Salary).reshape(-1,1)

#load the model

linear\_rmodel=linear\_model.LinearRegression()

#train the model : fit

linear\_rmodel.fit(x,y)

#test the model

# xtest=np.array([1.1,5.2,10.0]).reshape(-1,1)

xtest=x

ytest\_result=linear\_rmodel.predict(xtest)

print(xtest)

print(y)

print(ytest\_result)

#visualize the actual value v/s predicted value

plt.plot(x,y,color="red", label="Actual value")

plt.scatter(x,ytest\_result,color="blue", label="Pridicted value")

plt.xlabel("XTEST")

plt.ylabel("YTEST")

plt.title("Predicted value of Year of Experience")

plt.legend()

plt.show()

**Classification : Naive Byes Classifier**

from sklearn import model\_selection

import sklearn.datasets as db

from sklearn.naive\_bayes import GaussianNB

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

#Load the dataset

dataset = db.load\_iris()

#Identify X:sepal Length and Y: class

x = np.array(dataset.data[:, 0]).reshape(-1,1)

y = np.array(dataset.target).reshape(-1,1)

#Split the dataset into training & testing

xtrain, xtest, ytrain, ytest = model\_selection.train\_test\_split(x, y, test\_size=0.3)

#Load the classifier

classifier = GaussianNB()

#Train the model

classifier.fit(xtrain, ytrain)

#test the classifier

ytest\_predicted = classifier.predict(xtest)

print(xtest,ytest)

print(ytest\_predicted)

#Visualize the results

plt.scatter(xtest,ytest)

plt.scatter(xtest, ytest\_predicted)

plt.show()

**Heart Disease(Cholesterol)**

from ucimlrepo import fetch\_ucirepo

# fetch dataset

heart\_disease =fetch\_ucirepo(id=45)

# data (as pandas dataframes)

x = np.array(heart\_disease.data.features.chol).reshape(-1,1)

y = np.array(heart\_disease.data.targets ).reshape(-1,1)

# metadata

print(x)

# variable information

print(y)

#Split the dataset into training & testing

xtrain, xtest, ytrain, ytest = model\_selection.train\_test\_split(x, y, test\_size=0.3)

#Load the classifier

classifier = GaussianNB()

#Train the model

classifier.fit(xtrain, ytrain)

#test the classifier

ytest\_predicted = classifier.predict(xtest)

print(xtest,ytest)

print(ytest\_predicted)

#Visualize the results

plt.scatter(xtest,ytest)

plt.scatter(xtest, ytest\_predicted)

plt.show()

**Wine(Alcohol)**

from ucimlrepo import fetch\_ucirepo

# fetch dataset

wine =fetch\_ucirepo(id=109)

# data (as pandas dataframes)

x = np.array( wine.data.features.Alcohol).reshape(-1,1)

y = np.array(wine.data.targets ).reshape(-1,1)

# metadata

print(x)

# variable information

print(y)

#Split the dataset into training & testing

xtrain, xtest, ytrain, ytest = model\_selection.train\_test\_split(x, y, test\_size=0.3)

#Load the classifier

classifier = GaussianNB()

#Train the model

classifier.fit(xtrain, ytrain)

#test the classifier

ytest\_predicted = classifier.predict(xtest)

print(xtest,ytest)

print(ytest\_predicted)

#Visualize the results

plt.scatter(xtest,ytest)

plt.scatter(xtest, ytest\_predicted)

plt.show()