SVM

import pandas as pd

import matplotlib.pyplot as pit

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

from sklearn.svm import SVC

from sklearn.metrics import confusion\_matrix, accuracy\_score

dataset = pd.read\_csv("/content/drive/MyDrive/IRIS.csv")

x=dataset[["sepal\_length","sepal\_width","petal\_length","petal\_width"]].values

y=dataset["species"].values

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

Classifier = SVC(kernel="linear")

Classifier.fit(x\_train, y\_train)

y\_pred = Classifier.predict(x\_test)

accuracy = accuracy\_score(y\_test,y\_pred)\*100

confusion\_mat = confusion\_matrix(y\_test,y\_pred)

print("Accuracy for SVM is:",accuracy)

print("Confusion Matrix")

print(confusion\_mat)

Decision Tree

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

Iris\_data=pd.read\_csv("/content/drive/MyDrive/IRIS.csv")

plt.scatter(Iris\_data['sepal\_length'],Iris\_data['sepal\_width'])

plt.show()

from sklearn import tree

import graphviz

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split,cross\_val\_score

x=Iris\_data[["sepal\_length","sepal\_width","petal\_length","petal\_width"]]

y=Iris\_data["species"]

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.30,random\_state=42)

xt,xcv,yt,ycv=train\_test\_split(x\_train,y\_train,test\_size=0.10,random\_state=42)

Iris\_clf= DecisionTreeClassifier(criterion='gini',min\_samples\_split=2)

Iris\_clf.fit(xt,yt)

# print("Accuracy",cross\_val\_score(Iris\_clf,x\_train,y\_train,cv=1,scoring='accuracy'))

from sklearn.metrics import multilabel\_confusion\_matrix,accuracy\_score

Y\_h=Iris\_clf.predict(xcv)

# print("Accuracy:",accuracy\_score(ycv,Y\_h))

multilabel\_confusion\_matrix(ycv,Y\_h)

Bayes Theorem

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

from sklearn.preprocessing import StandardScaler

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

from sklearn.naive\_bayes import GaussianNB

df=pd.read\_csv("/content/drive/MyDrive/IRIS.csv")

x=df[["sepal\_length","sepal\_width","petal\_length","petal\_width"]].values

y=df["species"].values

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

sc=StandardScaler()

x\_train=sc.fit\_transform(x\_train)

x\_test=sc.transform(x\_test)

classifier = GaussianNB()

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

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

y\_pred

from sklearn import metrics

cnf\_metrix=metrics.confusion\_matrix(y\_test,y\_pred)

from sklearn.metrics import accuracy\_score

print("Accuracy :",accuracy\_score(y\_test,y\_pred))

cnf\_metrix

Logistic

from math import log

import matplotlib.pyplot as plt

import numpy as np

import statistics as st

import pandas as pd

from sklearn.linear\_model import LogisticRegression

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

from sklearn.metrics import mean\_squared\_error

from sklearn import preprocessing

df=pd.read\_csv("/content/drive/MyDrive/IRIS.csv")

le=preprocessing.LabelEncoder()

x=df[["sepal\_length","sepal\_width","petal\_length","petal\_width"]].values

y=df["species"].values

logReg=LogisticRegression()

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

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

y\_predict = logReg.predict(x\_test)

Linear

import matplotlib.pyplot as plt

import numpy as np

import statistics as st

import pandas as pd

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

df=pd.read\_csv("/content/drive/MyDrive/headbrain.csv")

x=df["Head Size(cm^3)"].values

y=df["Brain Weight(grams)"].values

m = len(x)

x=x.reshape(m,1)

reg=LinearRegression()

reg.fit(x,y)

print("Slope",reg.coef\_)

print("Intercept",reg.intercept\_)

y\_predict=reg.predict(x)

rmse=np.sqrt(mean\_squared\_error(y,y\_predict))

print("RMSE",rmse)

KNN

import matplotlib.pyplot as plt

import numpy as np

import statistics as st

import pandas as pd

from sklearn.neighbors import KNeighborsClassifier

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

from sklearn.metrics import mean\_squared\_error

from sklearn import preprocessing

df = pd.read\_csv("/content/drive/MyDrive/IRIS.csv")

x=df[["sepal\_length","sepal\_width","petal\_length","petal\_width"]].values

y=df["species"].values

knn = KNeighborsClassifier(n\_neighbors=3)

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

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

y\_predict = knn.predict(x\_test)

from sklearn import metrics

cnf\_metrix=metrics.confusion\_matrix(y\_test,y\_predict)

cnf\_metrix