# Important Libraries

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

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

data.head(10)

np.unique(data['species'], return\_counts=True)

y=x1w+x2+x3b

x=data.iloc[:,0].values

x

y=data.iloc[:,1].values

y

x=x.reshape(-1,1)

y=y.reshape(-1,1)

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

training\_x,testing\_x,training\_y,testing\_y=train\_test\_split(x,y,test\_size=0.2,random\_state=0)

training\_y

from sklearn.linear\_model import LinearRegression

Lin=LinearRegression()

Lin.fit(training\_x,training\_y)

pred\_y=Lin.predict(testing\_x)

print(pred\_y)

print(testing\_y)

pred\_y - testing\_y

plt.scatter(training\_x,training\_y,color='green')

plt.scatter(training\_x,training\_y,color='green')

plt.plot(training\_x,Lin.predict(training\_x),color='red')

Lin.coef\_

Lin.intercept\_

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

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

data

x=data.iloc[:,[0,1,2,3]].values

x

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

y

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

y=le.fit\_transform(y)

y

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

training\_x,testing\_x,training\_y,testing\_y=train\_test\_split(x,y,test\_size=0.3,random\_state=0)

training\_x

testing\_x

from sklearn.neighbors import KNeighborsClassifier

cls=KNeighborsClassifier()

cls.fit(training\_x,training\_y)

Pred\_y=cls.predict(testing\_x)

Pred\_y

testing\_y

from sklearn.metrics import confusion\_matrix,classification\_report

c\_n=confusion\_matrix(testing\_y,Pred\_y)

c\_n

print(classification\_report(testing\_y,Pred\_y))

from sklearn.neural\_network import MLPClassifier

mplc=MLPClassifier(hidden\_layer\_sizes=(50),max\_iter=500)

mplc.fit(training\_x,training\_y)

pred\_mplc=mplc.predict(testing\_x)

print(classification\_report(testing\_y,pred\_mplc))

from sklearn.svm import SVC

from sklearn import svm

clf=svm.SVC()

clf.fit(training\_x,training\_y)

pred\_mplc=mplc.predict(testing\_x)

print(classification\_report(testing\_y,pred\_mplc))

plt.subplot(2,1,1)

plt.plot(testing\_y)

plt.subplot(2,1,2)

plt.plot(pred\_mplc)