**KNN**

import pandas as pd

df=pd.read\_csv('t\_shirt\_size.csv')

df.head()

from sklearn.neighbors import KNeighborsClassifier

columns=['height','weight']

x=df[columns]

y=df['T\_shirt\_size']

model=KNeighborsClassifier(n\_neighbors=3)

model.fit(x,y)

predicted=model.predict([[161,61]])

predicted

**ANN**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

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

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

x = pd.DataFrame(df.iloc[:, 3:13].values)

y = df.iloc[:, 13].values

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

from sklearn.compose import ColumnTransformer

labelencoder=LabelEncoder()

x.loc[:, 2]= labelencoder.fit\_transform(x.iloc[:,2])

ct = ColumnTransformer ([("Country", OneHotEncoder(), [1])], remainder = "passthrough")

x = ct.fit\_transform(x)

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

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state = 0)

from sklearn.preprocessing import StandardScaler

sc= StandardScaler()

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

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

from keras.models import Sequential

from keras.layers import Dense

classifier=Sequential()

classifier.add(Dense(6, activation = 'relu', input\_dim= 10))

classifier.add(Dense (6, activation = 'relu'))

classifier.add(Dense (1, activation = "sigmoid"))

classifier.compile(optimizer = "adam", loss= 'binary\_crossentropy', metrics = ['accuracy'])

classifier.fit(x\_train, y\_train, batch\_size=12, epochs = 100)

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

y\_pred = (y\_pred > 0.5)

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

print(cm)

accuracy\_score (y\_test,y\_pred)

**KMEANS**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**from sklearn.cluster import KMeans**

**%matplotlib inline**

**X= -2 \* np.random.rand(100,2)**

**X1 = 1 + 2 \* np.random.rand(50,2)**

**X[50:100, :] = X1**

**plt.scatter(X[ : , 0], X[ :, 1], s = 50, c = 'b')**

**plt.show()**

**from sklearn.cluster import KMeans**

**Kmean = KMeans(n\_clusters=3)**

**Kmean.fit(X)**

**Kmean.cluster\_centers\_**

**plt.scatter(X[ : , 0], X[ : , 1], s =50, c='b')**

**plt.scatter(-0.94665068, -0.97138368, s=200, c='g', marker='s')**

**plt.scatter(2.01559419, 2.02597093, s=200, c='r', marker='s')**

**plt.show()**