DATASET:

* SUV.csv

FEATURES:

* User ID
* Gender
* Age
* Estimated Salary
* Purchased

SELECTED **X** :

* Age
* Estimated Salary

TARGET **Y** :

* Purchased or Not(0 | 1)

CODE :

# 1) APPLY KNN CLASSIFIER ON ANY DATASET AND EVALUATE THE PERFORMANCE

#DATA PREPROCESSING

import numpy as nm

import matplotlib.pyplot as mtp

import pandas as pd

data\_set= pd.read\_csv('/content/drive/MyDrive/datasets/suv\_data.csv')

#Extracting Independent and dependent Variable

x= data\_set.iloc[:, [2,3]].values

y= data\_set.iloc[:, 4].values

# Splitting the dataset into training and test set.

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

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

pak=x\_test

#label\_encoder object knows how to understand word labels.(on gender male=1,female=0)

from sklearn import preprocessing

label\_encoder = preprocessing.LabelEncoder()

data\_set['Gender']= label\_encoder.fit\_transform(data\_set['Gender'])

print(data\_set.head(),"\n")

#feature Scaling

from sklearn.preprocessing import StandardScaler

st\_x= StandardScaler()

x\_train= st\_x.fit\_transform(x\_train)

x\_test= st\_x.transform(x\_test)

#Fitting K-NN classifier to the training set

from sklearn.neighbors import KNeighborsClassifier

classifier= KNeighborsClassifier(n\_neighbors=5, metric='minkowski', p=2 )

classifier.fit(x\_train, y\_train)

#Predicting the test set result

y\_pred= classifier.predict(x\_test)

print(y\_pred)

#Creating the Confusion matrix

from sklearn.metrics import confusion\_matrix

print("\nPERFORMANCE METRIC[CONFUSION MATRIX]")

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

print(cm)

OUTPUT :

User ID Gender Age EstimatedSalary Purchased

0 15624510 1 19 19000 0

1 15810944 1 35 20000 0

2 15668575 0 26 43000 0

3 15603246 0 27 57000 0

4 15804002 1 19 76000 0

[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0

0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 0 0 1

0 0 0 0 1 1 1 1 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1]

PERFORMANCE METRIC[CONFUSION MATRIX]

[[64 4]

[ 3 29]]