**K-nearest neighbors algorithm**

**#Importing Packages**

import pandas as pd

from sklearn import preprocessing

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

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

from sklearn import neighbors

from sklearn import svm

**#Loading dataset**

dataset=pd.read\_csv(“train.csv”)

**#Converting text into numerical**

le=preprocessing.LabelEncoder()

le.fit(dataset[“Sex”])

Out[8]: LabelEncoder()

print(le.classes\_)

[‘female’ ‘male’]

**dataset[“Sex”] = le.transform(dataset[“Sex”])**

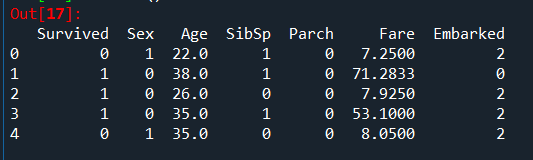
**dataset[“Embarked”] = le.fit\_transform(dataset[“Embarked”])**

**#Implementation of KNN**

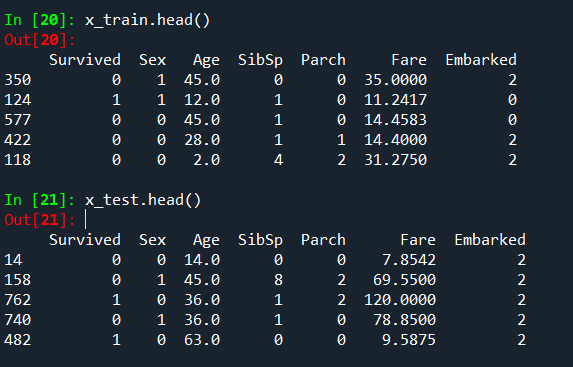
y=dataset[“Pclass”]

X=dataset.drop([“Pclass”, “PassengerId”,”Name”, “Cabin”, “Ticket”],axis=1)

X.head()



**x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3, random\_state = 0)**



def KNNA(k,X,y):

knn=neighbors.KneighborsClassifier(n\_neighbors=k)

score = knn.fit(x\_train, y\_train).score(x\_test,y\_test)

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

return score

for I in range(1,268):

score\_now=KNNA(I,X,y)

if(k\_val==0):

k\_val=i

acc\_score=score\_now

if(score\_now>acc\_score):

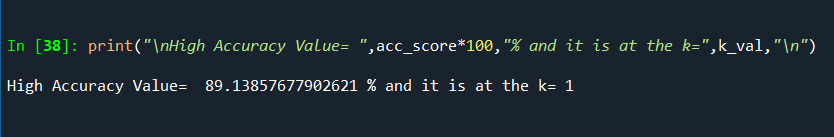
k\_val=i

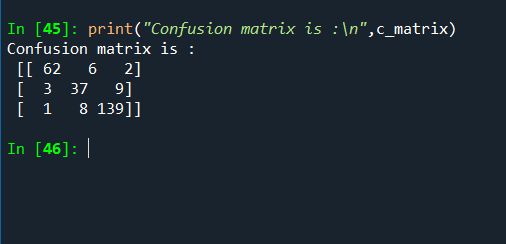
acc\_score=score\_now

knn = neighbors.KneighborsClassifier(n\_neighbors=k\_val)

y\_pred = knn.fit(x\_train,y\_train).predict(x\_test)

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





**Supoort Vector Machine**

clff = svm.SVC(gamma=0.01,C=100)

def Svm(X,y):

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.3,random\_state = 0)

y\_pred = clff.fit(X\_train,y\_train).predict(X\_test)

acc\_score = accuracy\_score(y\_test,y\_pred,normalize=True)

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

return acc\_score,conf\_mat

depVars = ['Survived','Pclass','SibSp','Sex','Parch','Embarked']

for i in depVars:

print(" ",i," vs Remaining:")

y = dataset[i]

X = dataset.drop(i,axis=1)

Score,Matrix = Svm(X,y)

print("Accuracy Score for", i ,"vs Remaining = ",Score\*100,"%\n")

print("Confusion Matrix is :-")

print(Matrix,"\n\n")

