**K\_nearest Neighbour**

**""" USER DEFINED FUNCTION"""**

**# function to find maximum positionum in list**

def Max(accuracy,k\_value,confusion):

# inbuilt function to find the position of maximum

maxpos = accuracy.index(max(accuracy))

max\_value\_cal.append(accuracy[maxpos])

max\_value\_cal.append(k\_value[maxpos])

max\_value\_cal.append(confusion[maxpos])

**# K\_neighbor function**

def K\_neighbors(DV,IDV):

from sklearn import neighbors

#spliting record of 70 training/ 30 testing

IDV\_train,IDV\_test,DV\_train,DV\_test=train\_test\_split(IDV,DV,test\_size=0.3,random\_state=0)

n=len(IDV\_test)

for i in range(n) :

knn=neighbors.KNeighborsClassifier(n\_neighbors=(i+1))

knn.fit(IDV\_train,DV\_train).score(IDV\_test,DV\_test)

DV\_pred=knn.predict(IDV\_test)

K\_value.append(i+1) # SELECTED K VALUE

Accuracy.append(knn.fit(IDV\_train,DV\_train).score(IDV\_test,DV\_test)) #accuracy value

Confusion.append(confusion\_matrix(DV\_test,DV\_pred)) # CONFUSION MATRIX ON SELECTING K VALUE

#print("x =",x)

if i==n:

break

**"""ENDS HERE """**

**“””IMPORTING DATASET “””**

import pandas as pd

K\_value = []

Accuracy = []

Confusion = []

max\_value\_cal = []

dataset=pd.read\_csv("train.csv")

dataset.columns

""" CPOLUMNS DISPLAYED

Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',

'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],

dtype='object')

"""

**""" IMporting all METHODS"""**

from sklearn import preprocessing

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

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

**""" preprocessing """**

#1)

le=preprocessing.LabelEncoder()

le.fit(dataset["Sex"])

""" Converting into numerical"""

dataset1=le.transform(dataset["Sex"])

dataset["Sex"]=dataset1

#2)

le=preprocessing.LabelEncoder()

le.fit(dataset["Embarked"])

""" Converting into numerical"""

dataset1=le.transform(dataset["Embarked"])

dataset["Embarked"]=dataset1

**""" Droping Uneccessary columns """**

Real\_data=dataset.drop(["Cabin","Name","PassengerId","Ticket"],axis=1)

**#LISTS to hold accuracy values and Confusion matrix values**

DV1=Real\_data["Pclass"]

#print(DV)

IDV1=Real\_data.drop("Pclass",axis=1)

#print(IDV)

K\_neighbors(DV1,IDV1)

Max(Accuracy,K\_value,Confusion)

**RESULT :**

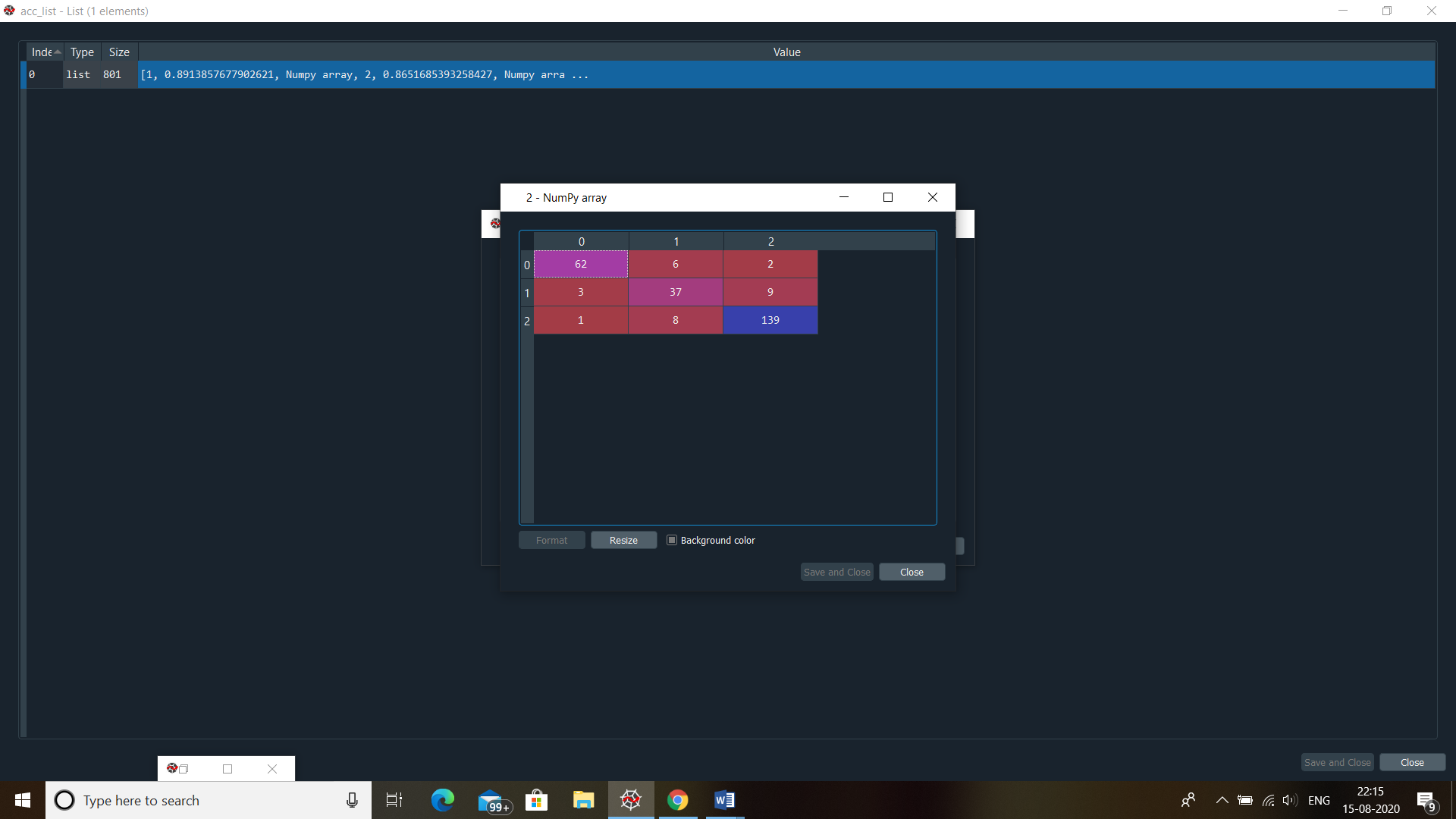
**Acc\_list hold all values which is in given format .first value is the k value and second value is the accuracy value , third is the confusion matrix.**

**Exp 1 :**

**K\_VALUE selected : 1**

**Accuracy Value : 0.8913857677902621**

**Confusion Matrix :**

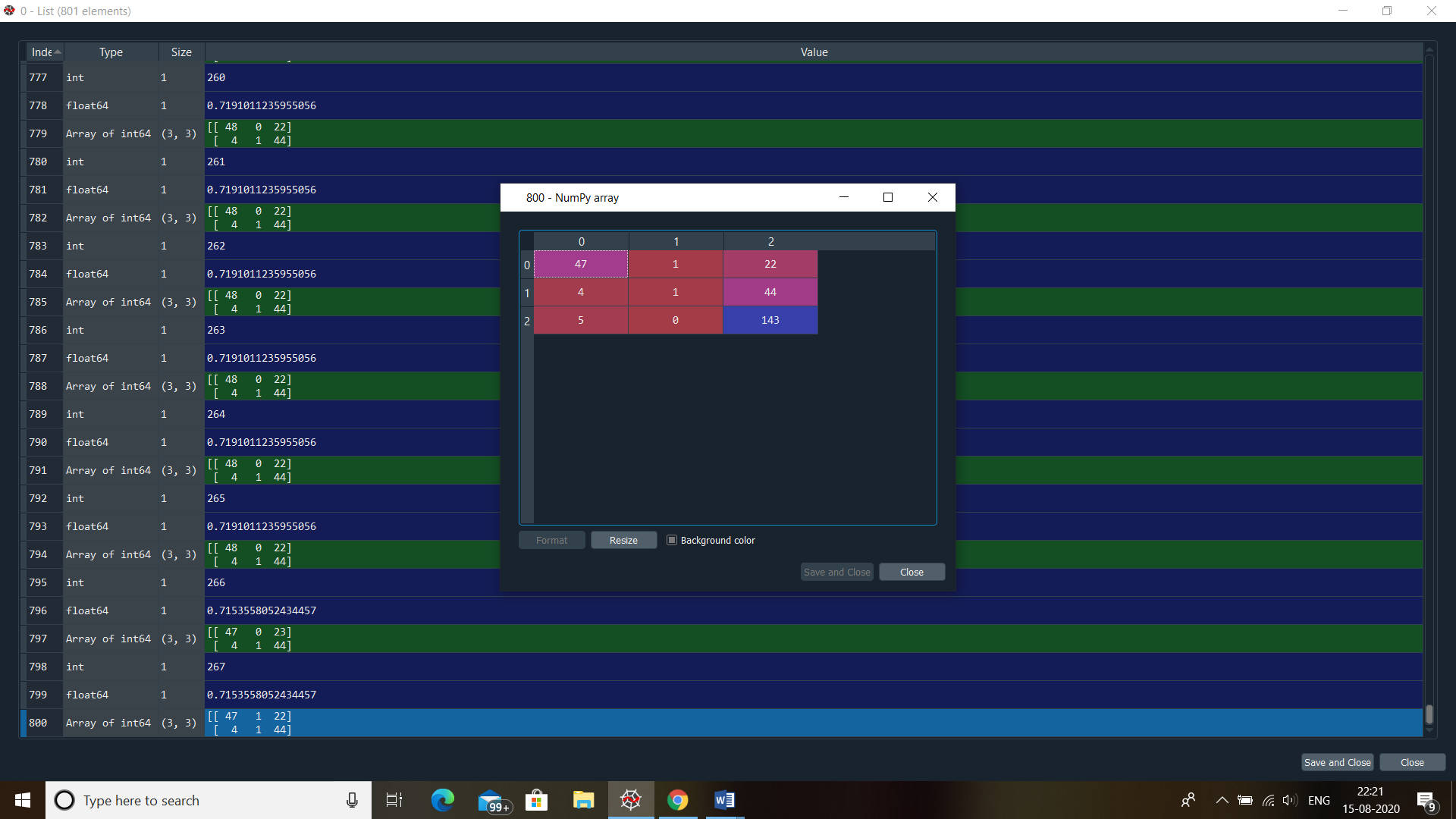


**Exp 2:**

**K\_VALUE selected : 267**

**Accuracy Value : 0.7153558052434457**

**Confusion Matrix :**



**Note :**

**Maximum accuracy achieved is 0.8913857677902621 at K value = 1 and its confusion matrix is**

