M.Nilofer Sultana, AP17110010058

splitting data into test and train, finding the neighbouring distance

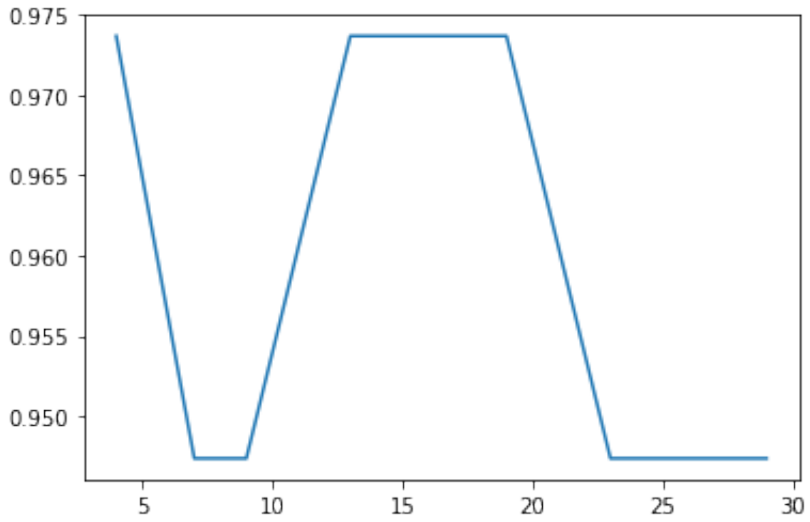
In [ ]:

```
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracyP_score
from sklearn.metrics import classification_report
from sklearn import datasets
import pandas as pd
import numpy as np
from sklearn import model_selection
import math
from sklearn.model_selection import train_test_split
iris=datasets.load_iris()
x=iris.data
y=iris.target
df=pd.DataFrame(x)
arr=np.array(df)
x_train,x_test,y_train,y_test=train_test_split(x,y)



trainx = np.array(x_train)
trainy = np.array(y_train)
testx = np.array(x_test)
testy = np.array(y_test)
print(testy[0])
s=0
mat = np.zeros([111,2],dtype=float)
for i in range(111):
    for j in range(4):
        s=s+math.sqrt((trainx[i][j]-testx[0][j])**2)
        if(j==3):
            mat[i][0]=s
            mat[i][1]=trainy[i]
            s=0
print(len(mat))
dic = {}
for i in range(len(mat)):
    dic[mat[i][0]]=mat[i][1]
print(dic)

for i in sorted (dic.keys()):
    #print(i,dic[i], end = "\n")
    mat=np.matrix([i,dic[i]])
    print(mat)

c0=0
c1=0
c2=0
k = int(input("Enter the value of k:"))
for i in range(k):
    if(mat.any()==0.0):
        c0+=1
    if(mat.any()==1.0):
        c1+=1
    if(mat.any()==2.0):
        c2+=1
print(a,b,c)
if (c0 >= c1) and (c0 >= c2):
   print("its versicolour")
elif (c1 >= c0) and (c1 >= c2):
   print("its setosa")
else:
   print("its virginica")
```

Confusion matrix and plotting the graph

In [8]:
```
from sklearn import neighbors,preprocessing
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix

scaler = preprocessing.StandardScaler().fit(x_train)
Xtrain = scaler.transform(x_train)
Xtest = scaler.transform(x_test)
a=[4,7,9,13,15,19,23,25,29]
b=[]
for i in range(len(a)):
    knn = neighbors.KNeighborsClassifier(n_neighbors=a[i])
    knn.fit(Xtrain, y_train)
    y_pred = knn.predict(Xtest)
    b.append(accuracy_score(y_test, y_pred))
print(b)
from matplotlib import pyplot as plt
plt.plot(a,b)
plt.show()
print(confusion_matrix(y_test, y_pred))
```

```
[0.9736842105263158, 0.9473684210526315, 0.9473684210526315, 0.97368
42105263158, 0.9736842105263158, 0.9736842105263158, 0.9473684210526
315, 0.9473684210526315, 0.9473684210526315]
```



```
[[12  0  0]
 [ 0 16  1]
 [ 0  1  8]]
```

In [ ]: