Day 27 - KNN

In [1]:

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  
  
dataset = pd.read\_csv("Dataset/train.csv")  
dataset = dataset.drop(["Name","Ticket","Cabin","PassengerId"],axis=1)

In [2]:

le=preprocessing.LabelEncoder()  
  
le.fit(dataset["Sex"])  
print(le.classes\_)  
dataset["Sex"] = le.transform(dataset["Sex"])  
  
le.fit(dataset["Embarked"])  
print(le.classes\_)  
dataset["Embarked"] = le.transform(dataset["Embarked"])

['female' 'male']  
['C' 'Q' 'S']

In [3]:

from sklearn import neighbors  
  
y=dataset["Pclass"]  
X=dataset.drop(["Pclass"],axis=1)  
  
X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.3,random\_state=0)  
  
total\_vals = y\_test.count()

### Defining custom function for k value iteration[¶](#Xc97e208bfa14925cee7afe220118e3e67adc2d1)

In [4]:

def accuracy\_fun(k):  
 knn=neighbors.KNeighborsClassifier(n\_neighbors=k)  
 return knn.fit(X\_train,y\_train).score(X\_test,y\_test)

### Applying the function in a for loop[¶](#Applying-the-function-in-a-for-loop)

In [5]:

lst = []  
  
for i in range(1, total\_vals+1):  
 lst.append(accuracy\_fun(i))

### Now getting the Max percentage value from the list with its k value[¶](#X53a8c4eb0c0263fb865a558577162dd712e7cf7)

In [6]:

maxpercent = max(lst)  
maxpos = lst.index(maxpercent)  
  
print("Maximum accuracy of, "+ str(round(maxpercent\*100,2)) +"% is obtained at k="+str(maxpos+1))

Maximum accuracy of, 89.14% is obtained at k=1

### Applying the k value obtained to create the confusion matrix.[¶](#X8520520a168ebbca722d90aa3e0efe37eb15225)

In [7]:

k=maxpos+1  
  
knn=neighbors.KNeighborsClassifier(n\_neighbors=k)  
knn.fit(X\_train,y\_train).score(X\_test,y\_test)

Out[7]:

0.8913857677902621

In [8]:

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

In [9]:

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

Out[9]:

array([[ 62, 6, 2],  
 [ 3, 37, 9],  
 [ 1, 8, 139]], dtype=int64)

In [10]:

import matplotlib.pyplot as plt   
   
x = [\*range(1,len(lst)+1)]   
y = lst   
   
plt.plot(x, y)   
   
plt.xlabel('K - value')   
plt.ylabel('Accuracy score')   
   
plt.title('k-value Vs Accuracy Score')   
  
plt.show()

![](data:image/png;base64;base64,)

In [ ]: