**DV🡪Pclass IDV🡪Survived,Sex,Age,SibSp,Parch,Fare,Embarked**

**Code:**

#import packages

import pandas as pd

import numpy as np

from sklearn import preprocessing

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

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

import matplotlib.pyplot as plt

#load dataset

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

#drop unimportant variables

data=data.drop(['PassengerId','Name','Ticket','Cabin'],axis=1)

#convert string to numerical values

le=preprocessing.LabelEncoder()

data["Sex"]=le.fit\_transform(data["Sex"])

data["Embarked"]=le.fit\_transform(data["Embarked"])

from sklearn import neighbors

#assign DV-->y and IDV-->X

y=data["Pclass"]

X=data.drop(["Pclass"],axis=1)

#split the dataset into train and test dataset

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

error = []

# Calculating error for K values between 1 and 267

for i in range(1,267):

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

knn.fit(X\_train, y\_train).score(X\_test,y\_test)

pred\_i = knn.predict(X\_test)

error.append(np.mean(pred\_i != y\_test))

#plot the graph

plt.figure(figsize=(12, 6))

plt.plot(range(1,267),error,color='red',linestyle='dashed',marker='o',markerfacecolor='blue', markersize=10)

plt.title('Error Rate K Value')

plt.xlabel('K Value')

plt.ylabel('Mean Error')

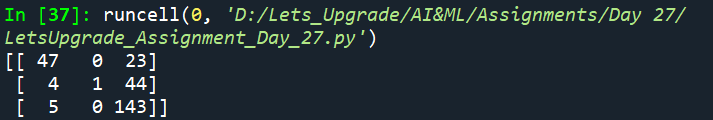
#prediction

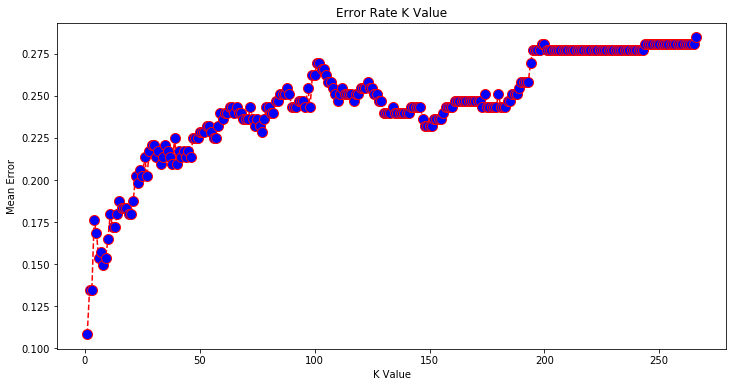
y\_pred=knn.predict(X\_test)

#confusion matrix

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

**Output:**





Consider k=1 for more accuracy as it has least error rate