**KNN**

**Import 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 svm

import numpy as np

from sklearn import neighbors

**Import data:**

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

dataset.columns

del dataset["Ticket"]

del dataset["Cabin"]

del dataset["Name"]

del dataset["PassengerId"]

**Convert catagorical to numeric:**

le=preprocessing.LabelEncoder()

le.fit(dataset["Sex"])

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

dataset["Fare"]=np.round(dataset["Fare"])

dataset["Age"]=np.round(dataset["Age"])

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

**Split data x and y:**

y=dataset["Survived"]

x=dataset.drop(["Survived"],axis=1)

**Split the data:**

xtrain,xtest,ytrain,ytest=train\_test\_split(x,y,test\_size=0.3,random\_state=0)

knn=neighbors.KNeighborsClassifier(n\_neighbors=3)

knn.fit(xtrain,ytrain)

ypre=knn.predict(xtest)

print(accuracy\_score(ytest, ypre,normalize=True))

print(confusion\_matrix(ytest, ypre))

0.6853932584269663

[[123 34]

[ 50 60]]

**Using forloop for k value:**

for i in range(25,40):

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

knn.fit(xtrain,ytrain);

ypre=knn.predict(xtest);

print(accuracy\_score(ytest, ypre,normalize=True));

print(confusion\_matrix(ytest, ypre))

0.6779026217228464

[[134 23]

[ 63 47]]

0.6591760299625468

[[135 22]

[ 69 41]]

0.6704119850187266

[[135 22]

[ 66 44]]

0.6591760299625468

[[136 21]

[ 70 40]]

0.6591760299625468

[[136 21]

[ 70 40]]

0.6591760299625468

[[136 21]

[ 70 40]]

0.6554307116104869

[[134 23]

[ 69 41]]

0.6554307116104869

[[135 22]

[ 70 40]]

0.6591760299625468

[[135 22]

[ 69 41]]

0.6591760299625468

[[135 22]

[ 69 41]]

0.6554307116104869

[[133 24]

[ 68 42]]

0.6704119850187266

[[137 20]

[ 68 42]]

0.6629213483146067

[[134 23]

[ 67 43]]

0.6666666666666666

[[136 21]

[ 68 42]]

0.6591760299625468

[[134 23]

[ 68 42]]

Here higher accuracy is 0.6779026217228464