**Naïve bayes**

**Import packages:**

import pandas as pd

from sklearn import preprocessing

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

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

import numpy as np

**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"])

**Find classes:**

le.fit(dataset["Sex"])

print(le.classes\_)

[0 1]

**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)

from sklearn.naive\_bayes import \*

clf=BernoulliNB()

y\_pre=clf.fit(xtrain,ytrain).predict(xtest)

**Accuracy:**

accuracy\_score(ytest, y\_pre,normalize=True)

confusion\_matrix(ytest, y\_pre)

Out[106]:

array([[131, 26],

[ 35, 75]], dtype=int64)

**Accuracy of all catagorical variable useing for loop**:

def accuracy\_var(y,x):

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

clf=BernoulliNB()

y\_pre=clf.fit(xtrain,ytrain).predict(xtest)

print(accuracy\_score(ytest, y\_pre,normalize=True))

print(confusion\_matrix(ytest, y\_pre))

k=list(["Pclass","Survived","Sex","SibSp","Parch","Embarked"])

for i in k:

y=pd.DataFrame(dataset[i]);

x=dataset.drop([i],axis=1);

accuracy\_var(y,x)

0.5917602996254682

[[ 30 8 32]

[ 9 10 30]

[ 24 6 118]]

0.7715355805243446

[[131 26]

[ 35 75]]

0.7453183520599251

[[ 49 49]

[ 19 150]]

0.6891385767790262

[[162 20 0 0 0 0 0]

[ 43 22 0 0 0 0 0]

[ 6 2 0 0 0 0 0]

[ 5 2 0 0 0 0 0]

[ 2 0 0 0 0 0 0]

[ 1 0 0 0 0 0 0]

[ 2 0 0 0 0 0 0]]

0.7153558052434457

[[182 16 0 0 0]

[ 31 9 0 0 0]

[ 24 3 0 0 0]

[ 1 0 0 0 0]

[ 1 0 0 0 0]]

0.7340823970037453

[[ 0 0 49]

[ 0 0 22]

[ 0 0 196]]

Here survived accuracy level higher to compare others.