

# ML (Naive Bayes Classifier)

# Roll No.: 19103014

# Name: Amritpal Singh

# Branch: CSE

# Group: G1

#!/usr/bin/env python

# coding: utf-8

# In[263]:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import math

# In[264]:

train\_data = pd.read\_csv('Titanic\_train.csv')

test\_data = pd.read\_csv('Titanic\_test.csv')

submissions = pd.read\_csv('Titanic\_gender\_submission.csv')

del test\_data['Name']

del test\_data['PassengerId']

del test\_data['Ticket']

del test\_data['Cabin']

train\_data['Sex'] = train\_data['Sex'].replace(["female", "male"], [0, 1])

train\_data['Age'] = pd.qcut(train\_data['Age'], 10, labels=False)

train\_data['Fare'] = pd.qcut(train\_data['Fare'], 10, labels=False)

test\_data['Sex'] = test\_data['Sex'].replace(["female", "male"], [0, 1])

test\_data['Age'] = pd.qcut(test\_data['Age'], 10, labels=False)

test\_data['Fare'] = pd.qcut(test\_data['Fare'], 10, labels=False)

test\_data = test\_data.values

features = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']

# In[265]:

y\_counts=train\_data['Survived'].value\_counts()

p\_y=(y\_counts/len(train\_data)).to\_dict()

# In[266]:

features\_prob = []

for i in features:

count = train\_data[i].value\_counts()

features\_prob.append((count/len(train\_data)).to\_dict())

# In[267]:

df\_survived=train\_data.loc[train\_data['Survived'] == 1]

df\_died=train\_data.loc[train\_data['Survived'] == 0]

# In[268]:

conditional\_prob\_survived = []

for i in features:

count = df\_survived[i].value\_counts()

conditional\_prob\_survived.append((count/len(df\_survived)).to\_dict())

# In[269]:

conditional\_prob\_died = []

for i in features:

count = df\_died[i].value\_counts()

conditional\_prob\_died.append((count/len(df\_died)).to\_dict())

# In[ ]:

# In[270]:

def Bayes(row):

res\_survived = p\_y[0]

num = 1

den = 1

for i in range(len(row)):

if math.isnan(row[i]):

continue

try:

num\*= conditional\_prob\_survived[i][row[i]]

den\*= features\_prob[i][row[i]]

except KeyError:

continue

res\_survived\*= num

res\_survived/=den

res\_died = p\_y[0]

num = 1

den = 1

for i in range(len(row)):

if math.isnan(row[i]):

continue

try:

num\*= conditional\_prob\_died[i][row[i]]

den\*= features\_prob[i][row[i]]

except KeyError:

continue

res\_died\*= num

res\_died/=den

if res\_survived>=res\_died:

return 1

else:

return 0

# In[271]:

Output = []

for row in test\_data:

Output.append(Bayes(row))

# In[272]:

x = submissions['Survived'].values

accuracy = 0

for i in range(len(x)):

if(x[i]==Output[i]):

accuracy+=1

print('Accuracy : ',accuracy/len(x) \* 100)

OUTPUT :

