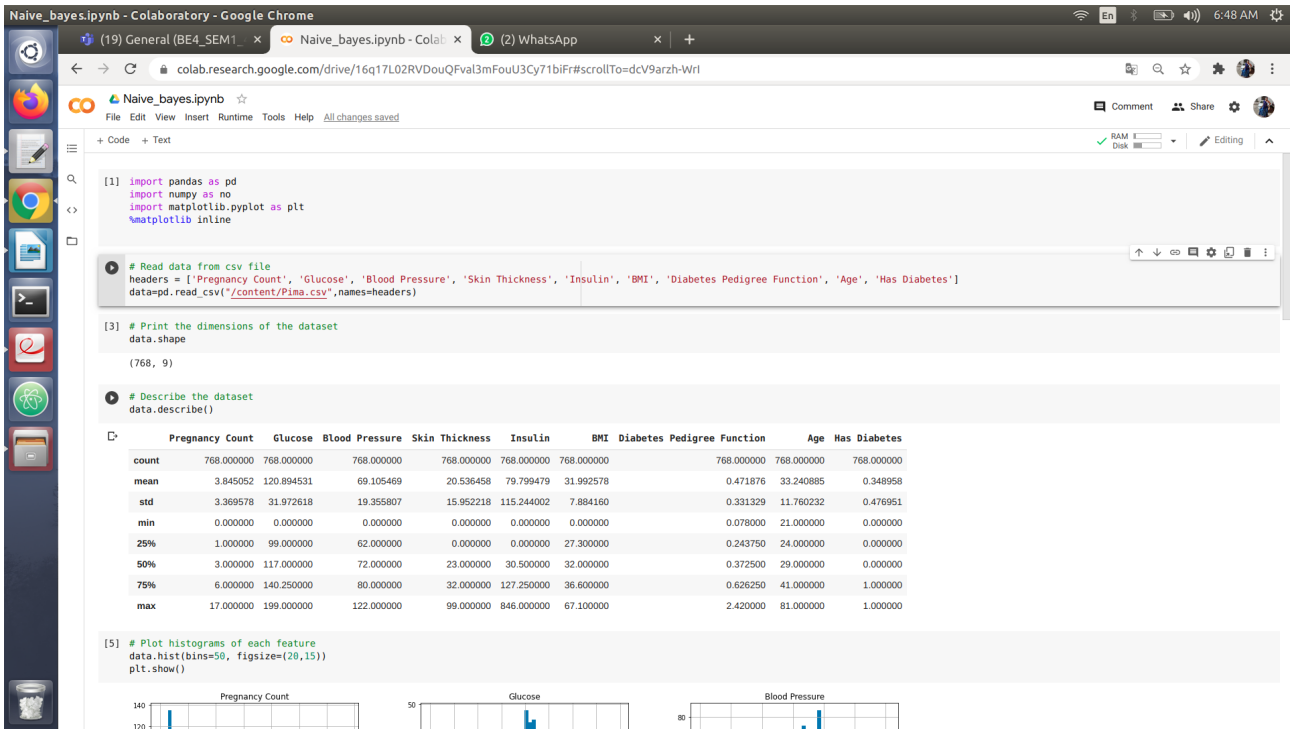
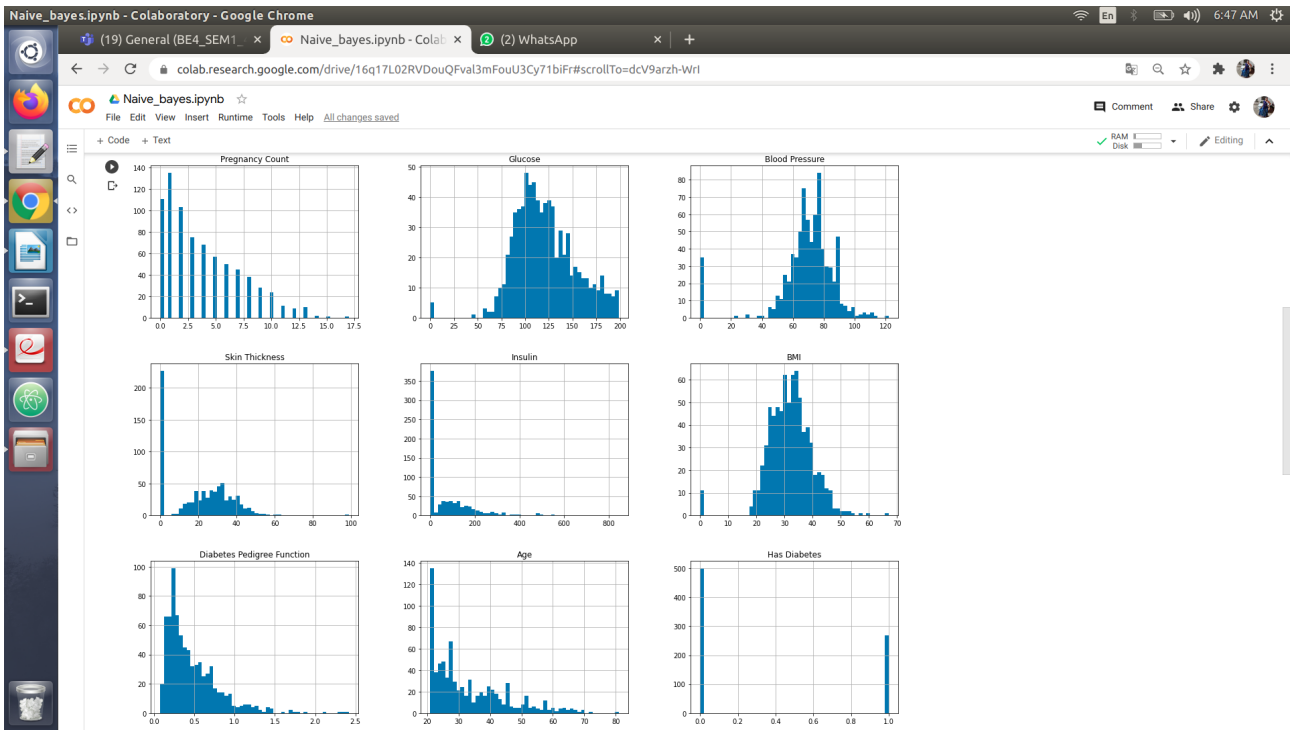
CODE:

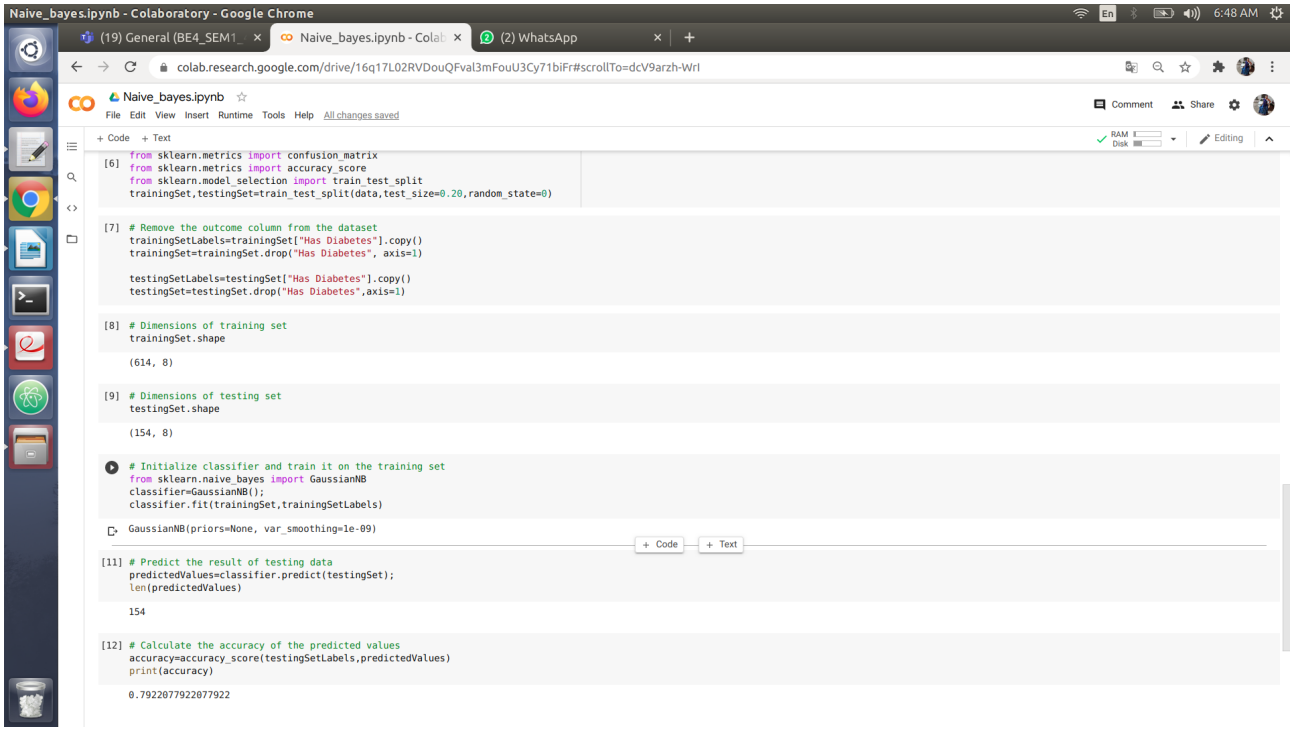
```
# -*- coding: utf-8 -*-
"""Naive_bayes.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/16q17L02RVDouQFval3mFouU3Cy71biFr
"""

# Commented out IPython magic to ensure Python compatibility.
import pandas as pd
import numpy as no
import matplotlib.pyplot as plt
# %matplotlib inline

# Read data from csv file
headers = ['Pregnancy Count', 'Glucose', 'Blood Pressure', 'Skin Thickness', 'Insulin', 'BMI',
'Diabetes Pedigree Function', 'Age', 'Has Diabetes']
data=pd.read_csv("/content/Pima.csv",names=headers)

# Print the dimensions of the dataset
data.shape

# Describe the dataset
data.describe()

# Plot histograms of each feature
data.hist(bins=50, figsize=(20,15))
plt.show()

# Seperate the dataset into training and testing set (80% training and 20% testing)
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
trainingSet,testingSet=train_test_split(data,test_size=0.20,random_state=0)

# Remove the outcome column from the dataset
trainingSetLabels=trainingSet["Has Diabetes"].copy()
trainingSet=trainingSet.drop("Has Diabetes", axis=1)

testingSetLabels=testingSet["Has Diabetes"].copy()
testingSet=testingSet.drop("Has Diabetes",axis=1)

# Dimensions of training set
trainingSet.shape

# Dimensions of testing set
testingSet.shape

# Initialize classifier and train it on the training set
```

```
from sklearn.naive_bayes import GaussianNB
classifier=GaussianNB();
classifier.fit(trainingSet,trainingSetLabels)

# Predict the result of testing data
predictedValues=classifier.predict(testingSet);
len(predictedValues)

# Calculate the accuracy of the predicted values
accuracy=accuracy_score(testingSetLabels,predictedValues)
print(accuracy)
```

OUTPUT:




Naive_bayes.ipynb - Colaboratory - Google Chrome
6:47 AM
(19) General (BE4_SEM1_
Naive_bayes.ipynb - Colab
(2) WhatsApp
colab.research.google.com/drive/16q17L02RVDouQFval3mFouU3Cy71biFr#scrollTo=dcV9arzh-WrI
Naive_bayes.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Pregnancy Count
Glucose
Blood Pressure
Skin Thickness
Insulin
BMI
Diabetes Pedigree Function
Age
Has Diabetes


Naive_bayes.ipynb - Colaboratory - Google Chrome
6:48 AM
(19) General (BE4_SEM1_
Naive_bayes.ipynb - Colab
(2) WhatsApp
colab.research.google.com/drive/16q17L02RVDouQFval3mFouU3Cy71biFr#scrollTo=dcV9arzh-WrI
Naive_bayes.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[6] from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
trainingSet,testingSet=train_test_split(data,test_size=0.20,random_state=0)
[7] # Remove the outcome column from the dataset
trainingSetLabels=trainingSet["Has Diabetes"].copy()
trainingSet=trainingSet.drop("Has Diabetes", axis=1)
testingSetLabels=testingSet["Has Diabetes"].copy()
testingSet=testingSet.drop("Has Diabetes",axis=1)
[8] # Dimensions of training set
trainingSet.shape
(614, 8)
[9] # Dimensions of testing set
testingSet.shape
(154, 8)
# Initialize classifier and train it on the training set
from sklearn.naive_bayes import GaussianNB
classifier=GaussianNB();
classifier.fit(trainingSet,trainingSetLabels)
GaussianNB(priors=None, var_smoothing=1e-09)
+ Code + Text
[11] # Predict the result of testing data
predictedValues=classifier.predict(testingSet);
len(predictedValues)
154
[12] # Calculate the accuracy of the predicted values
accuracy=accuracy_score(testingSetLabels,predictedValues)
print(accuracy)
0.7922077922077922