**ASSIGNMENT 2**

**Problem Statement**:

Download Pima Indians Diabetes dataset. Use Naïve Bayes‟ Algorithm for classification” .Load the data from CSV file and split it into training and test datasets. Summarize the properties in the training dataset so that we can calculate probabilities and make predictions. Classify samples from a test dataset and a summarized training dataset.

1. Create your own data set (min 15 to 20 tuples) and also import diabetes data set

2. Normalize and preprocess data (like fill in missing values or delete tuples with missing values, categorization of data if required)

3. Using R/Python read training and Test data, split data into training and testing

4. Apply Naive Bayes model and predict accuracy, confusion matrix.

5. Analyze result Analyze result for different test samples (70:30 ratio, 60:40 ratio or

Completely new test sample etc).

Analyze result for different attribute selection.

Analyze result with missing values and without missing values. Plot graph accordingly

**Theory**:

It is a classification technique based on [Bayes’ Theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem" \t "_blank) with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability. Naive Bayes model is easy to build and particularly useful for very large data sets.

Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

The fundamental Naive Bayes assumption is that each feature makes an:

* independent
* equal

Contribution to the outcome.

With relation to our dataset, this concept can be understood as:

* We assume that no pair of features are dependent. For example, the temperature being ‘Hot’ has nothing to do with the humidity or the outlook being ‘Rainy’ has no effect on the winds. Hence, the features are assumed to be **independent**.
* Secondly, each feature is given the same weight (or importance). For example, knowing only temperature and humidity alone can’t predict the outcome accurately. None of the attributes is irrelevant and assumed to be contributing **equally** to the outcome.

**Code:**

importnumpy as np

import pandas as pd

importmatplotlib.pyplot as plt

fromsklearn.model\_selection import train\_test\_split

fromsklearn.metrics import accuracy\_score

fromsklearn.metrics import confusion\_matrix

fromsklearn.naive\_bayes import GaussianNB

df=pd.read\_csv('PimaIndiansDiabetes.csv')

df.info()

df.head()

df.describe()

print(df.describe().to\_string())

plt.figure(figsize=(14,8))

df.boxplot()

plt.show()

x=df[["TimesPregnant","GlucoseConcentration","BloodPrs","SkinThickness","Serum","BMI","DiabetesFunct","Age"]]

y=df["Class"]

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=42)

model=GaussianNB()

model.fit(X\_train,Y\_train)

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

accuracy=accuracy\_score(Y\_test,y\_pred)

print(accuracy)

cm=confusion\_matrix(Y\_test,y\_pred)

print(cm)

print(df['Serum'].mean())

df['Serum']=df['Serum'].replace(0,df['Serum'].mean())

x=df[["TimesPregnant","GlucoseConcentration","SkinThickness","Serum","BMI","DiabetesFunct","Age"]]

y=df["Class"]

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=42)

model=GaussianNB()

model.fit(X\_train,Y\_train)

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

accuracy=accuracy\_score(Y\_test,y\_pred)

print(accuracy)

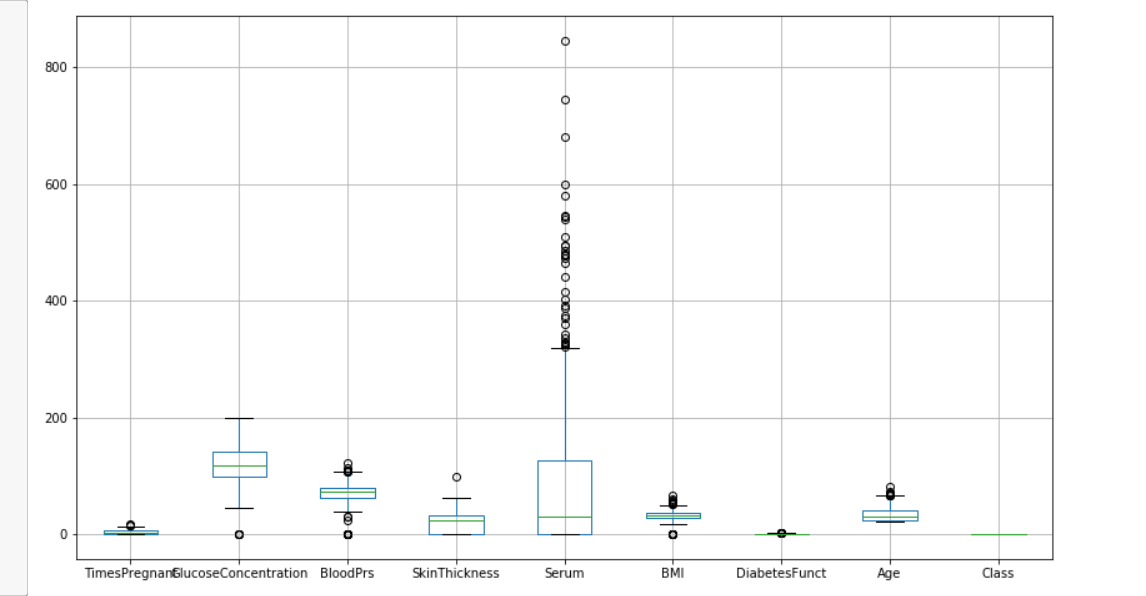
cm=confusion\_matrix(Y\_test,y\_pred)

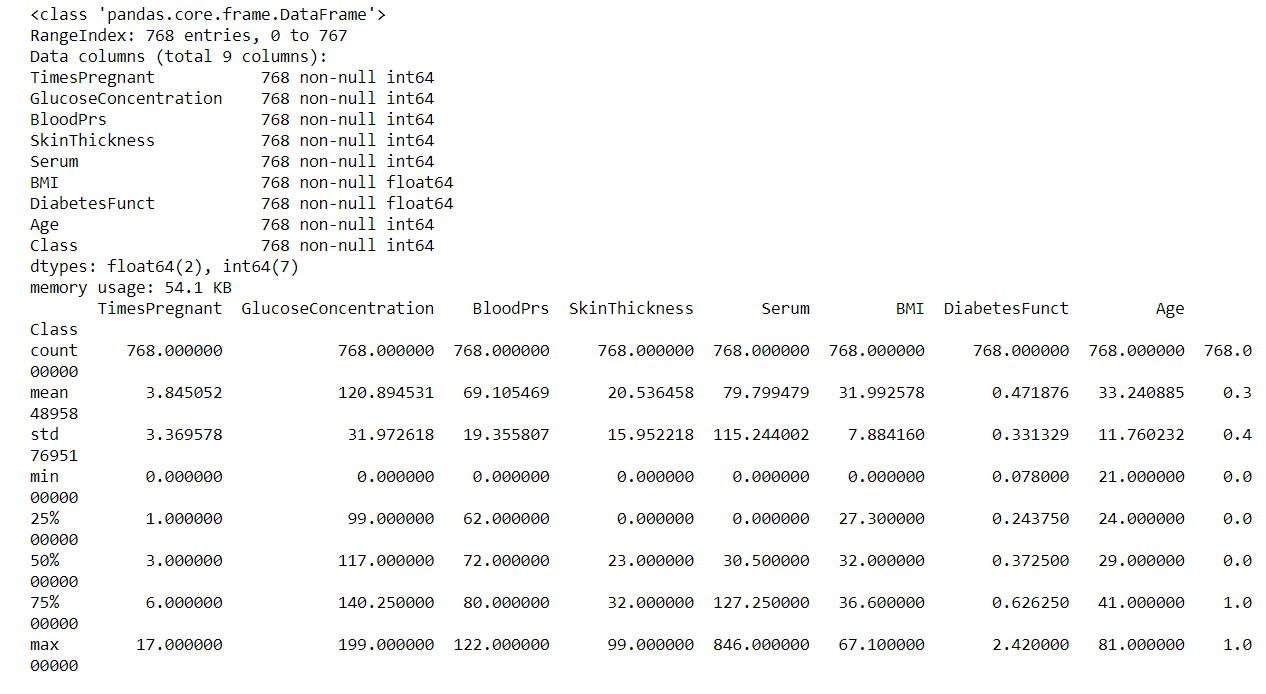
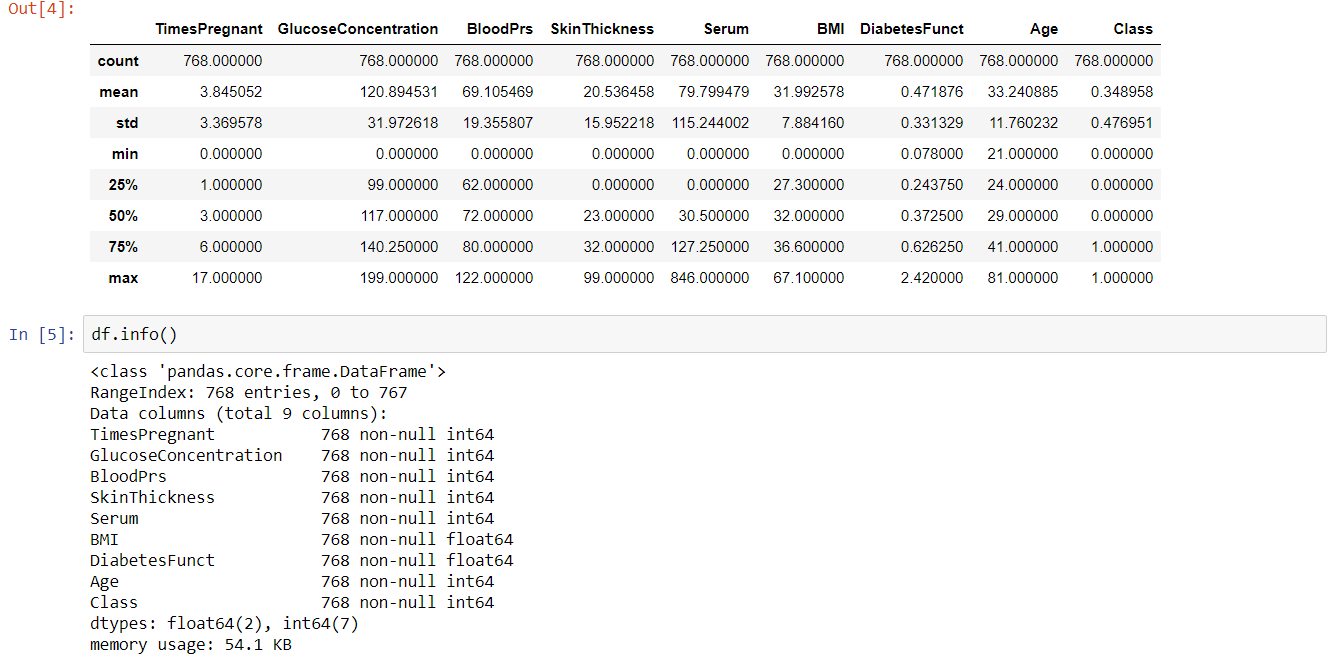
print(cm)

**Dataset**:

Pima Indian Diabetes

1. Times Pregnant
2. Glucose Concentration
3. BloodPrs
4. SkinThickness
5. Serum
6. BMI
7. Diabetes Funct
8. Age
9. Class

**Output**: 



**Conclusion**: Thus we have normalized and summarized the Pima Indian diabetes dataset and performed preprocessing on the dataset.