**Experiment No.05**

**A.1 Aim:** Implementation of Naïve Bayes Algorithm using any programming language like JAVA, C++, Python or WEKA Tool.

**PART B**

(PART B: TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)***

|  |  |
| --- | --- |
| **Roll No.** 50 | **Name:** AMEY THAKUR |
| **Class:** Comps TE B | **Batch:** B3 |
| **Date of Experiment:** 15/04/2021 | **Date of Submission:** 15/04/2021 |
| **Grade:** |  |

**B.1 Software Code written by a student:**

***(Paste your problem statement related to your case study completed during the 2 hours of practice in the lab here)***

import numpy as np

import pandas as pd

# Import necessary modules

from sklearn.naive\_bayes import GaussianNB

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

d1 = pd. read\_csv ('diabetes\_csv.csv')

d1.head()

# Loading data

# Create feature and target arrays

X = d1[d1.columns[:-1]]

y = d1[d1.columns[-1]]

# Split into training and test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

test\_size = 0.2, random\_state=42)

gnb = GaussianNB()

gnb.fit(X\_train, y\_train)

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

d1

from sklearn import metrics

print("Gaussian Naive Bayes model accuracy(in %):",

metrics.accuracy\_score(y\_test, y\_pred)\*100)

# Predict on dataset which model has not seen before

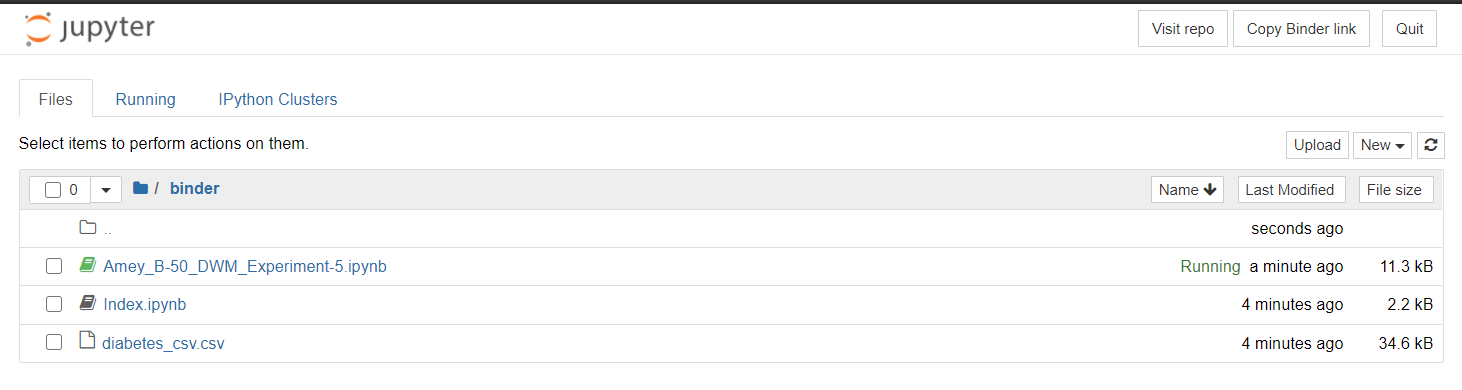
print(gnb.predict(X\_test))

**B.2 Input and Output:**

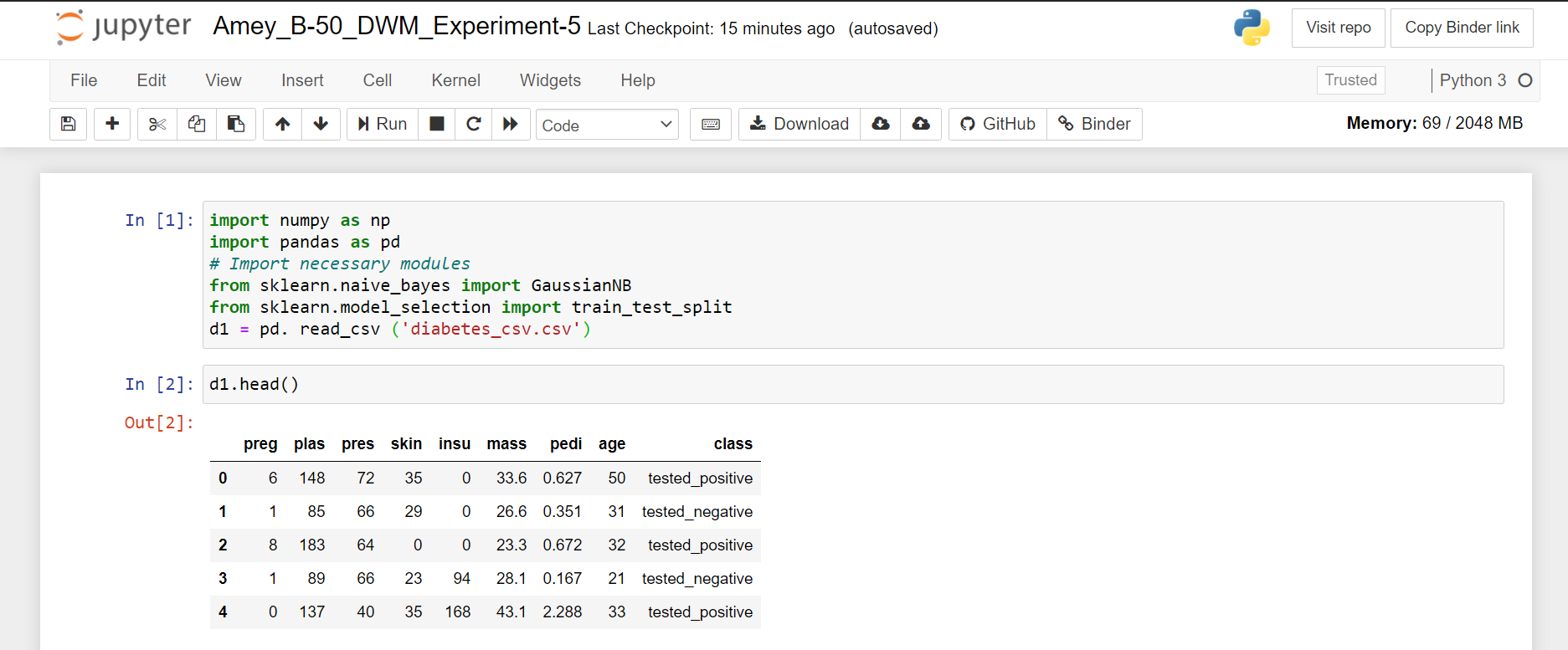
***(Paste your program input and output in the following format, If there is an error then paste the specific error in the output part. In case of an error with the due permission of the faculty, an extension can be given to submit the error-free code with output in due course of time. Students will be graded accordingly.)***

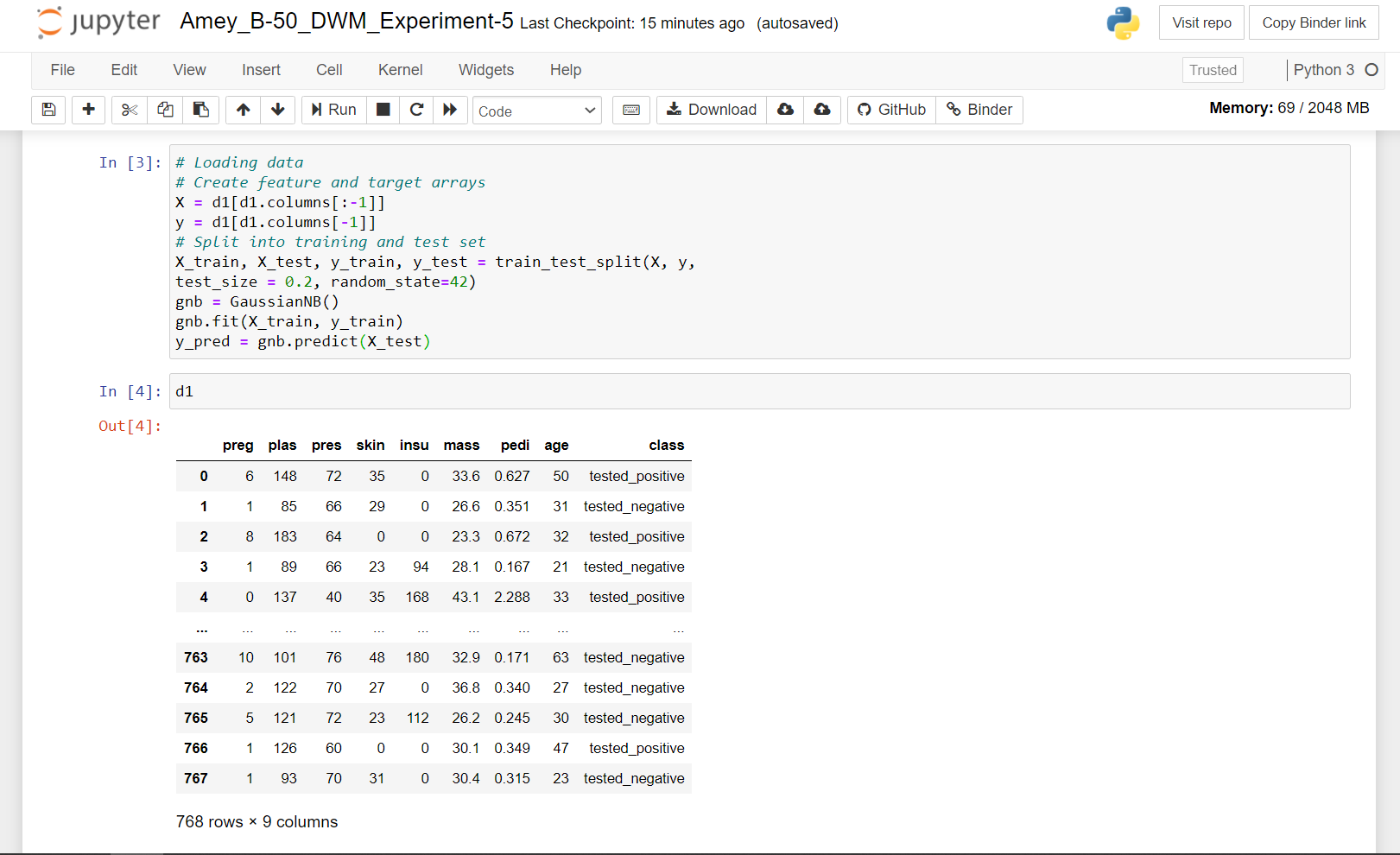
**Jupyter Notebook**

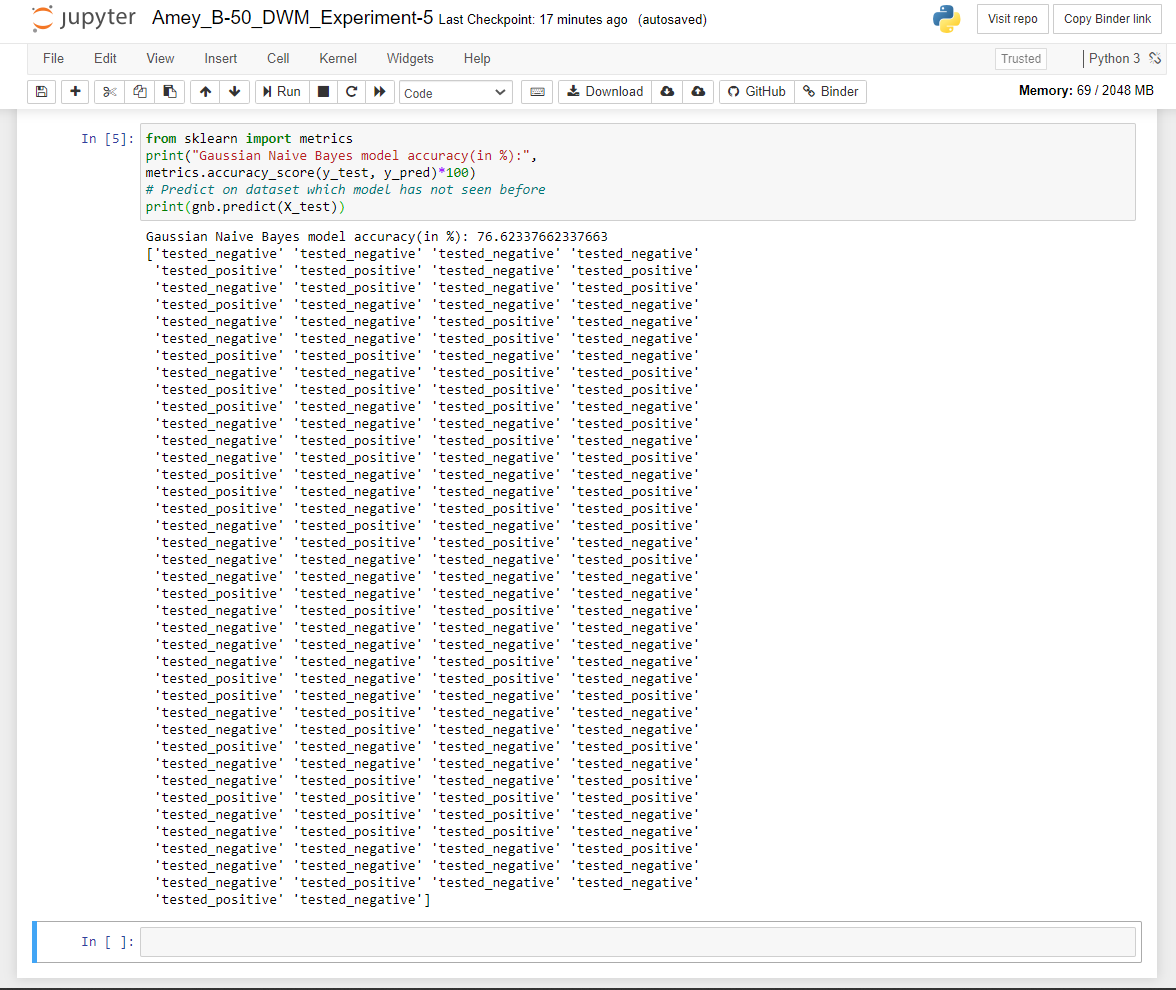
**Jupyter/Binder:**

****

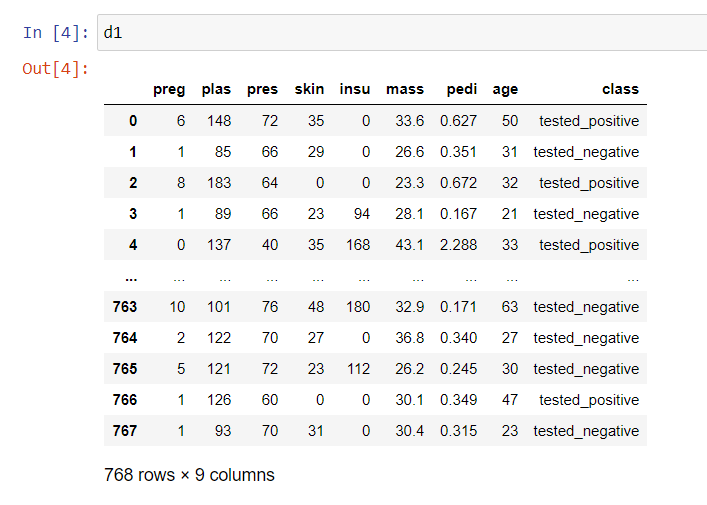
**Jupyter Notebook:**

****

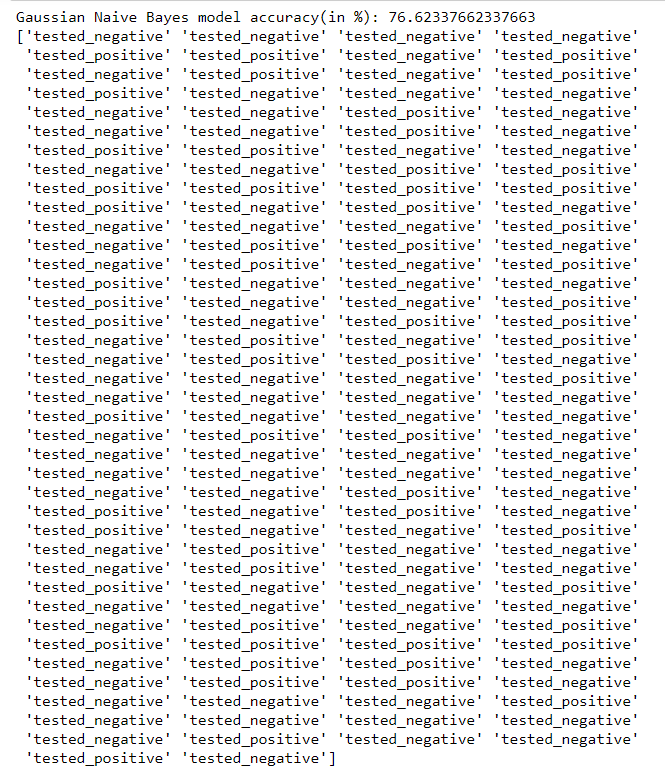
****

****

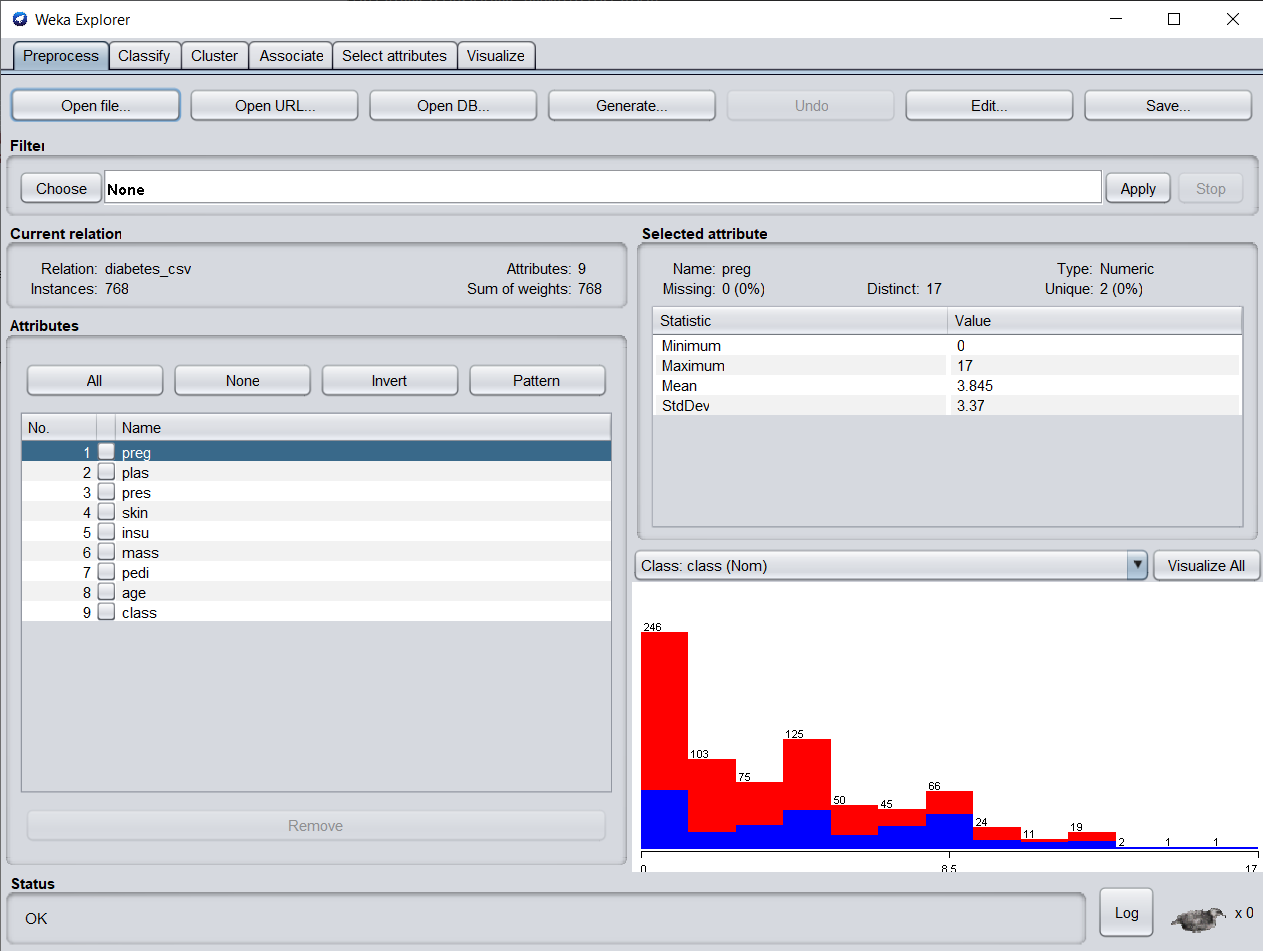
**Sample Dataset:**

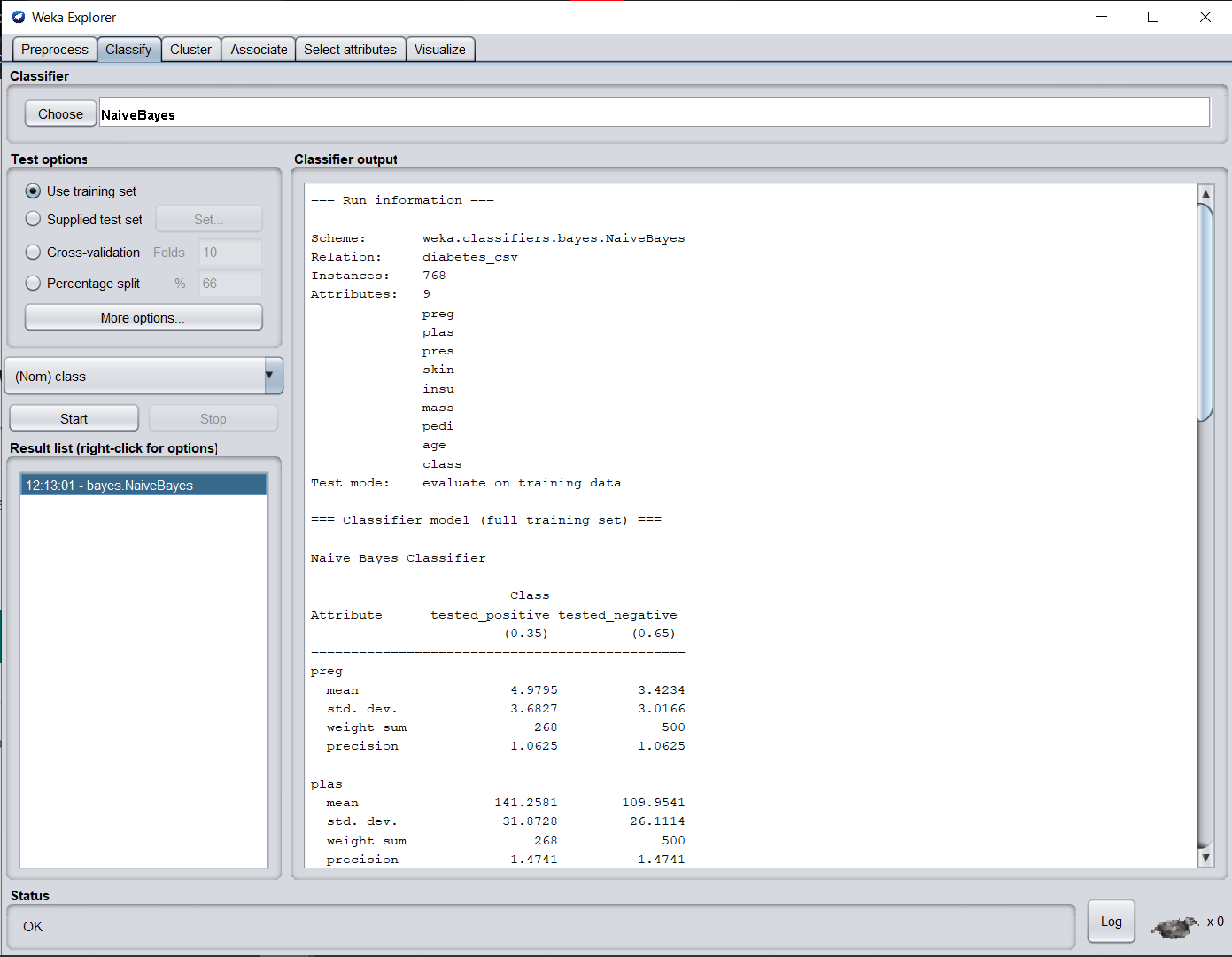
****

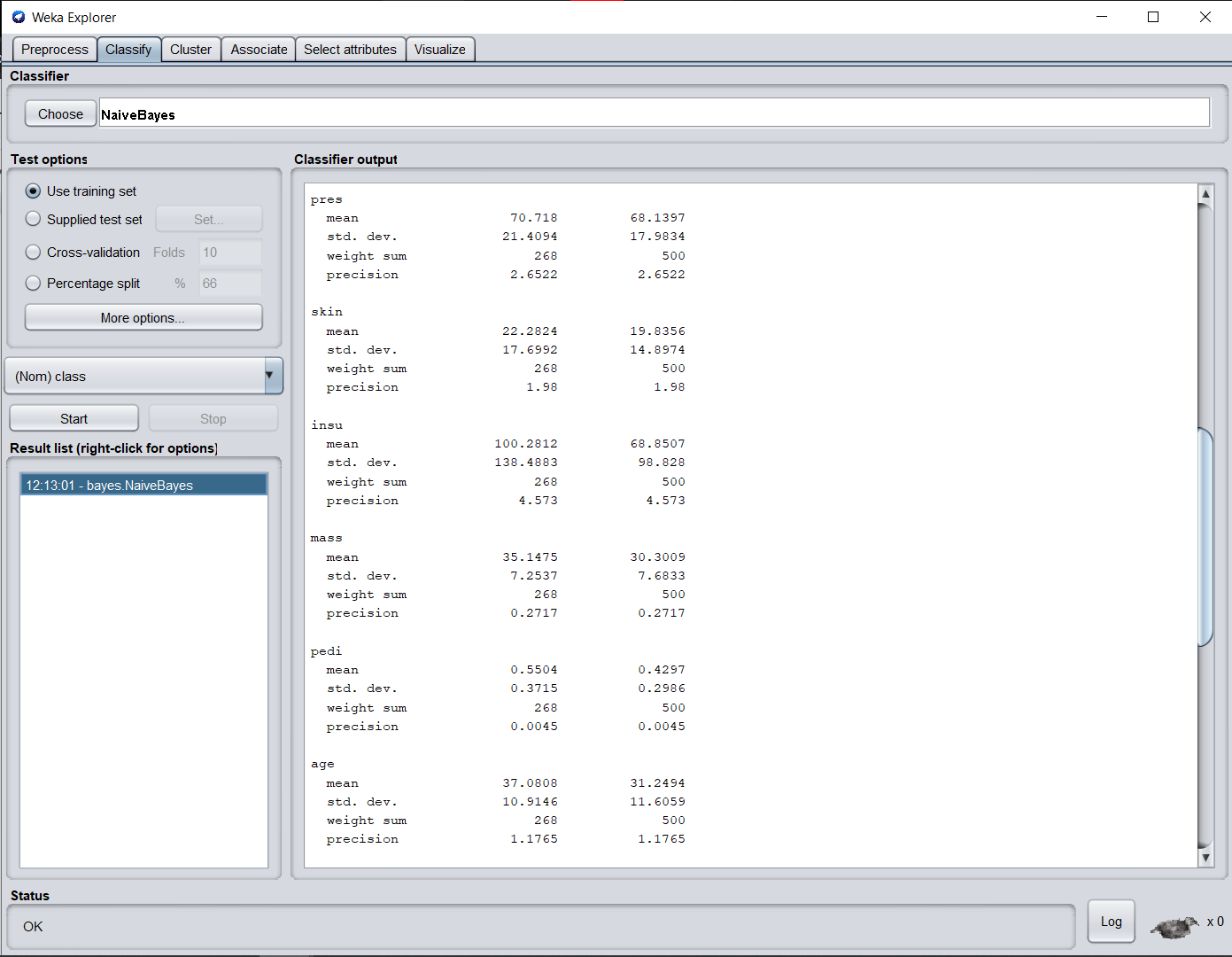
**Python Output:**

****

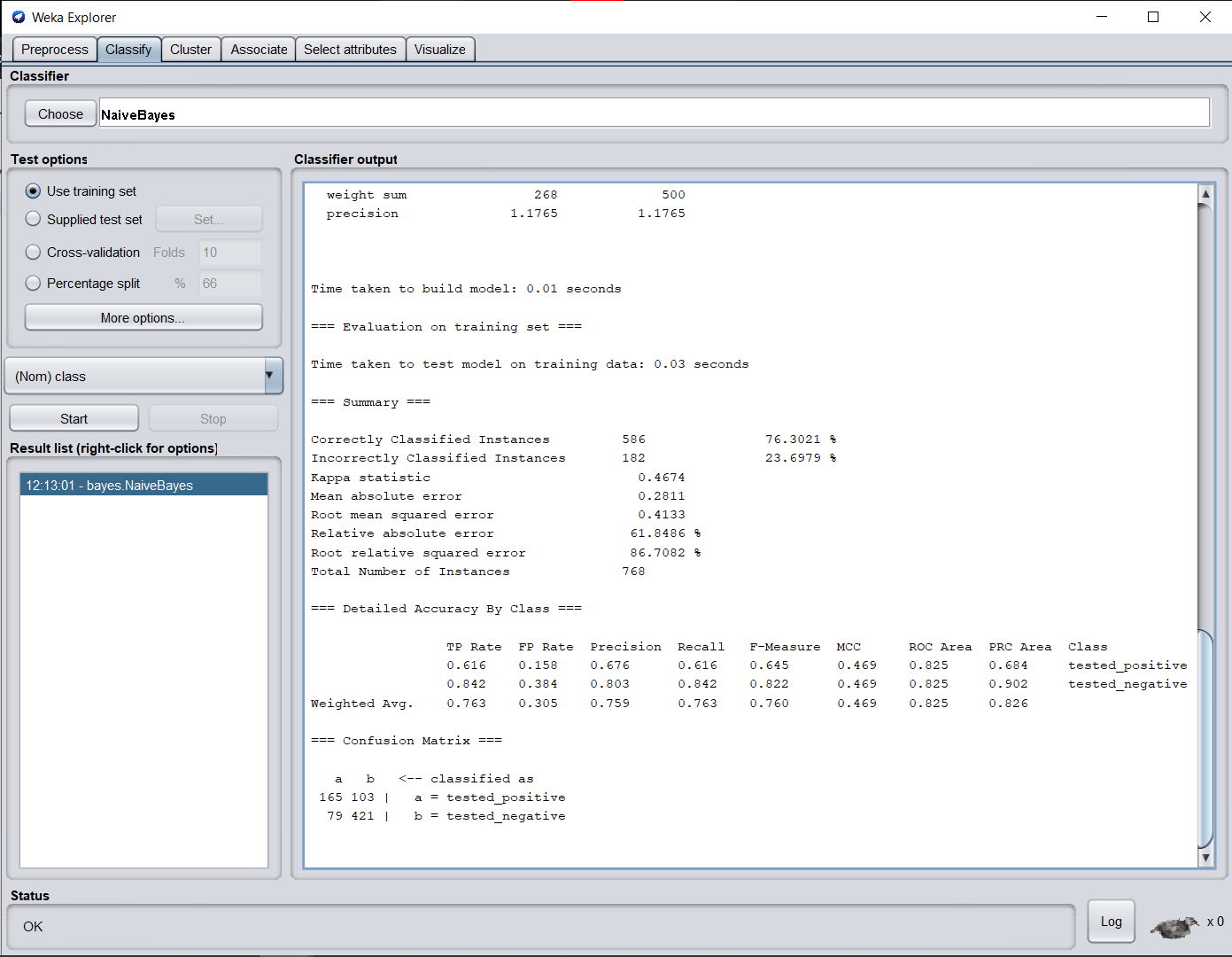
**Weka Tool**

****

****

****

**Weka Output:**

****

**Naive Bayes model accuracy (in %):**

76.62337662337663 (Using Python)

76.3021 (Using Weka)

**B.3 Observations and learning:**

***(Students are expected to comment on the output obtained with clear observations and learning for each task/ subpart assigned)***

We observed that Bayesian Classification represents a supervised learning method as well as a statistical classification method. The Bayesian classification is used as a probabilistic learning method as Naive Bayes classifiers are among the most successful known algorithms for learning to classify the datasets. Naïve Bayes classification Algorithm is one of the probabilistic algorithms which classifies the datasets according to its knowledge data and creates the Result as per the given knowledge.

**B.4 Conclusion:**

*(****Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)***

* Naïve Bayes classification Algorithm is one of the probabilistic algorithms which classify the datasets according to their knowledge data and creates the Result as per the given knowledge.
* Hence we’ve successfully implemented the Naive Bayesian Algorithm through Python as well as Weka Tool.

**B.5 Question of Curiosity**

***(To be answered by the student based on the practical performed and learning/observations)***

1. List out other classifiers and their efficiency.

**Ans:**

1. Decision trees
2. Bayesian classifiers
3. Neural networks
4. Nearest neighbour classifiers
5. Support vector machines
6. Linear regression.

Comparison of different classifier techniques with efficiency:

DT=decision tree,

NB=Naive Bayes classifier,

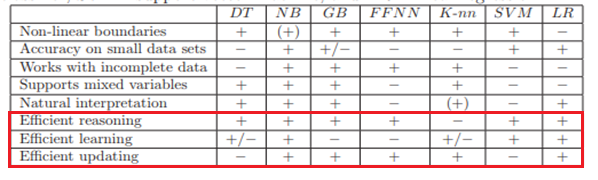
GB=general Bayesian classifier,

F F NN= feed-forward neural network,

K-nn=K-nearest neighbour classifier,

SV M=support vector machine, and

LR= linear regression.

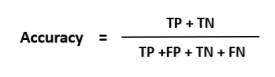


1. What is the accuracy of the classifier explain with an example?

**Ans:**

The accuracy of a classifier is given as the percentage of total correct predictions divided by the total number of instances.

Mathematically,



If the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data tuples for which the class label is not known.