**Experiment No. 4**

Aim: To implement Bayesian algorithm.

Requirements: Windows O.S, Weka tool, Python and Python libraries: Pandas, Numpy, Sklearn and Matplot.

Problem Statement: To implement Naïve Bayes algorithm on iris data set using Weka tool and Python.

Theory:

Naive Bayes: Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. Bayes’ theorem states the following relationship, given class variable y and dependent feature vector x1 through xn, :

P(y∣x1,…,xn) = P(y)P(x1,…,xn∣y)/P(x1,…,xn)

Using the naive conditional independence assumption that

P(xi|y,x1,…,xi−1,xi+1,…,xn) = P(xi|y),

for all i, this relationship is simplified to

P(y∣x1,…,xn) = P(y)∏i=1nP(xi∣y)/P(x1,…,xn)

Since P(x1,…,xn) is constant given the input, we can use the following classification rule:

P(y∣x1,…,xn) ∝ P(y)∏i=1nP(xi∣y)⇓

y^ = argmax(y)(P(y)∏i=1nP(xi∣y)),

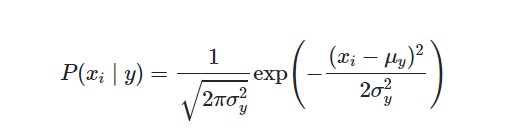
and we can use Maximum A Posteriori (MAP) estimation to estimate P(y) and P(xi∣y); the former is then the relative frequency of class y in the training set.

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of P(xi∣y).

In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. (For theoretical reasons why naive Bayes works well, and on which types of data it does, see the references below.)

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

Gaussian Naive Bayes: When working with continuous data, an assumption often taken is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. The likelihood of the features is assumed to be-

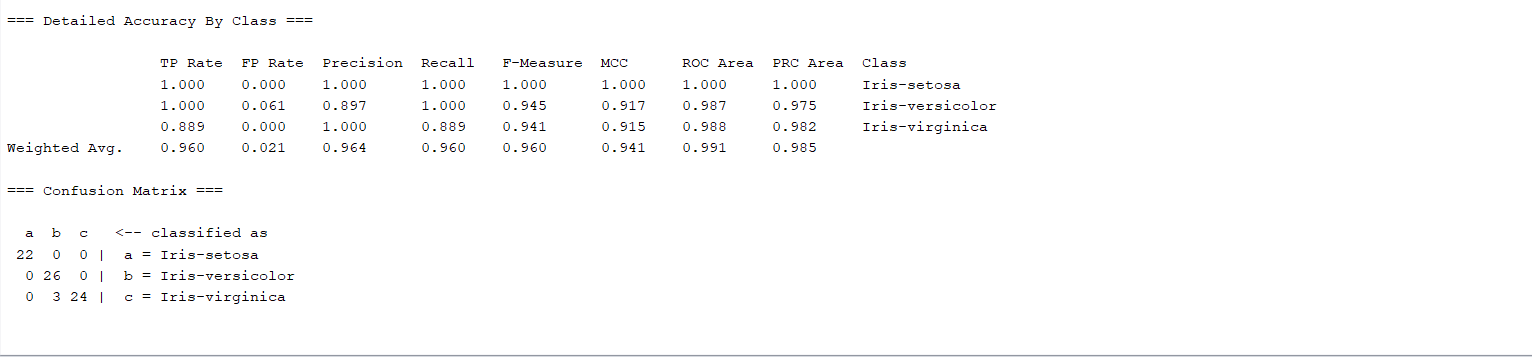
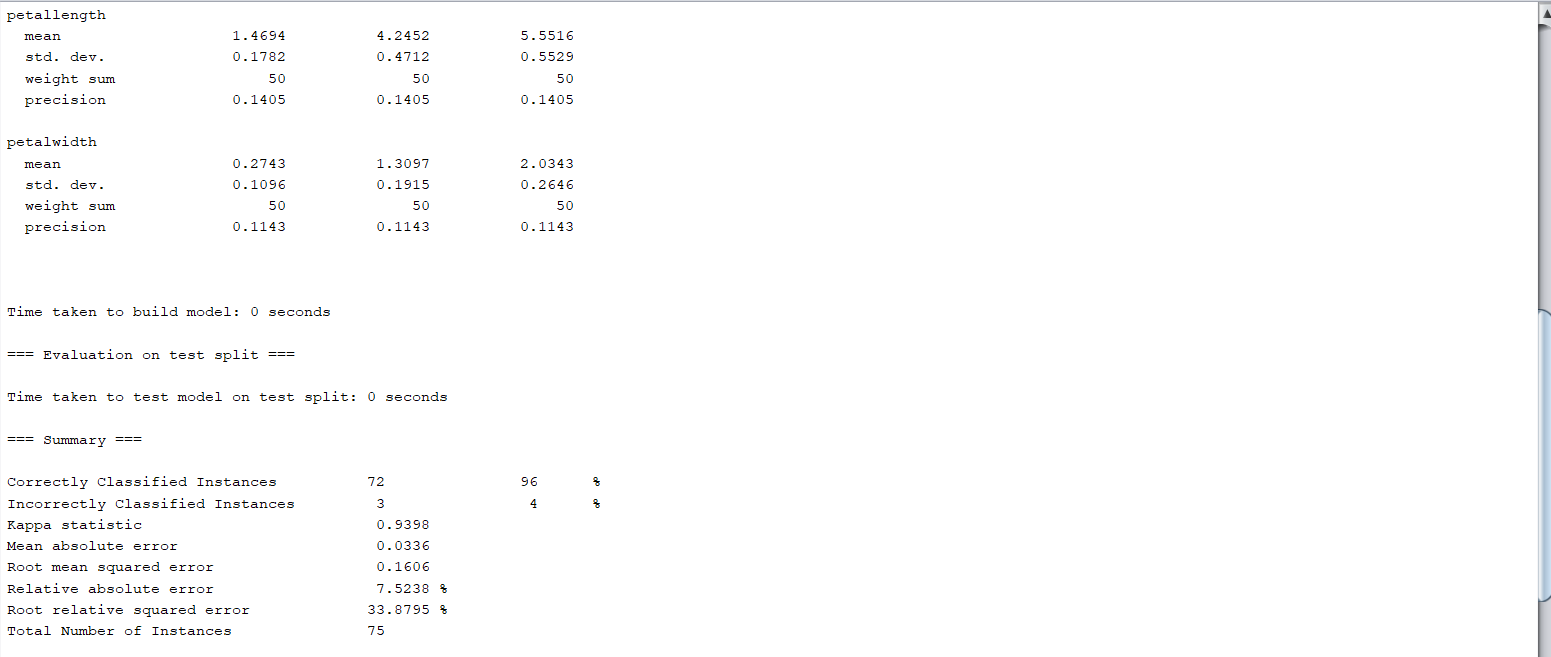
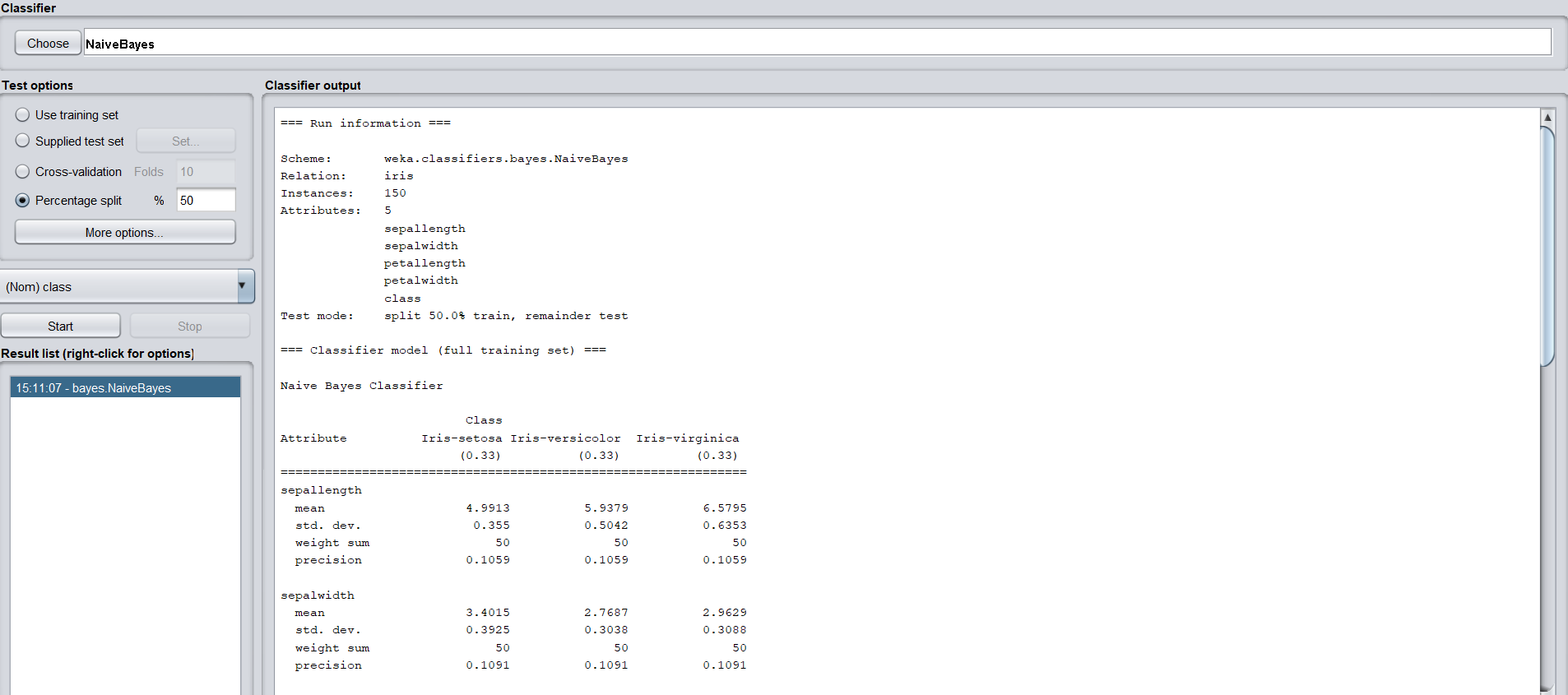
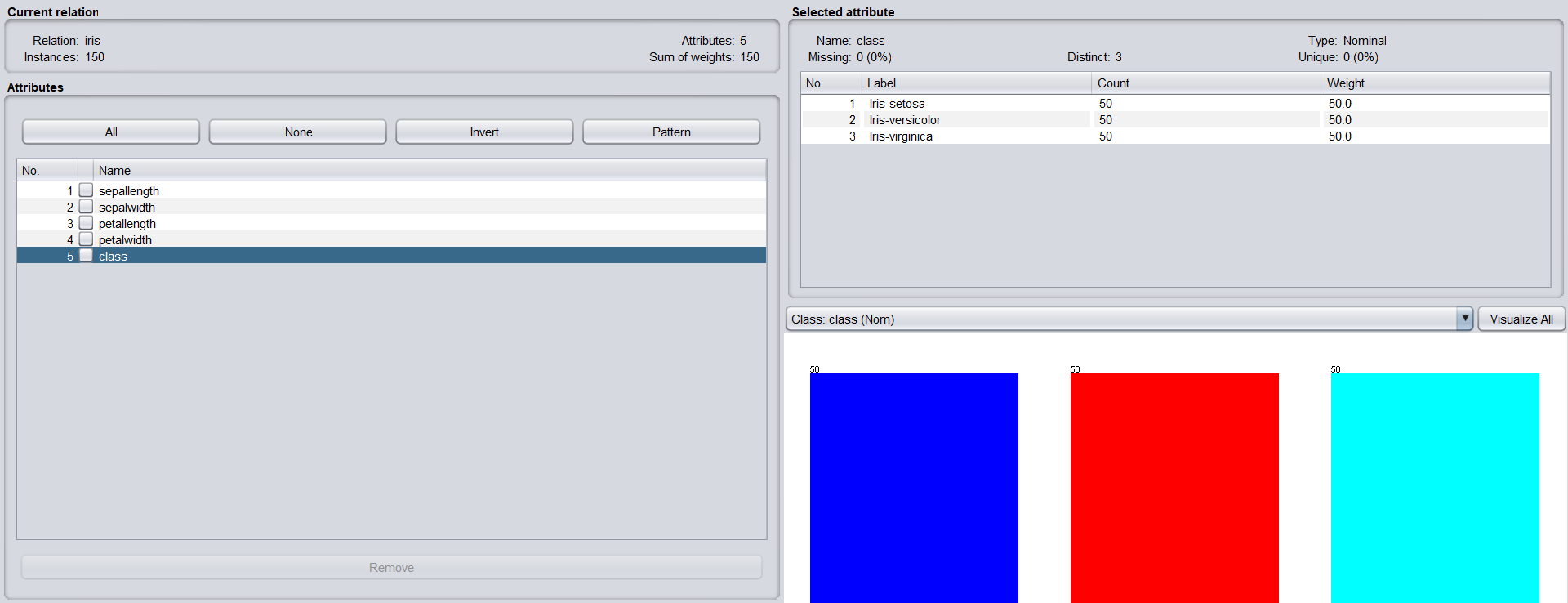
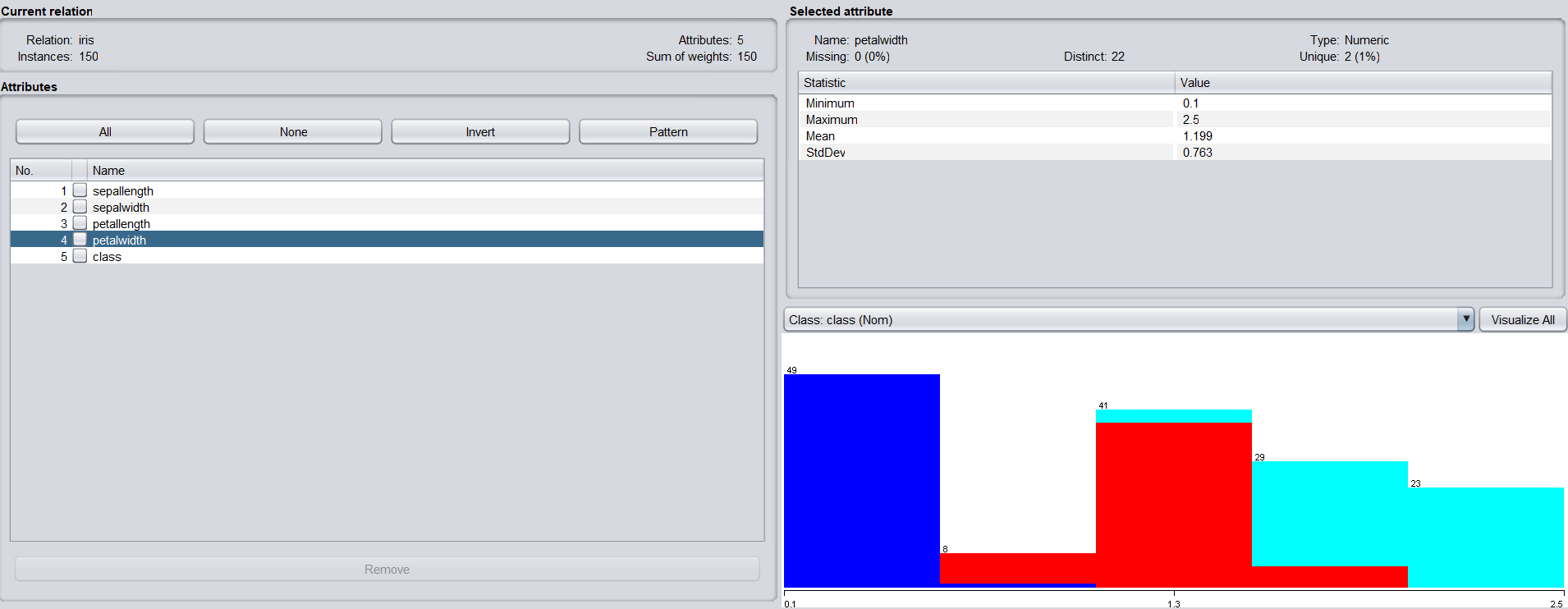
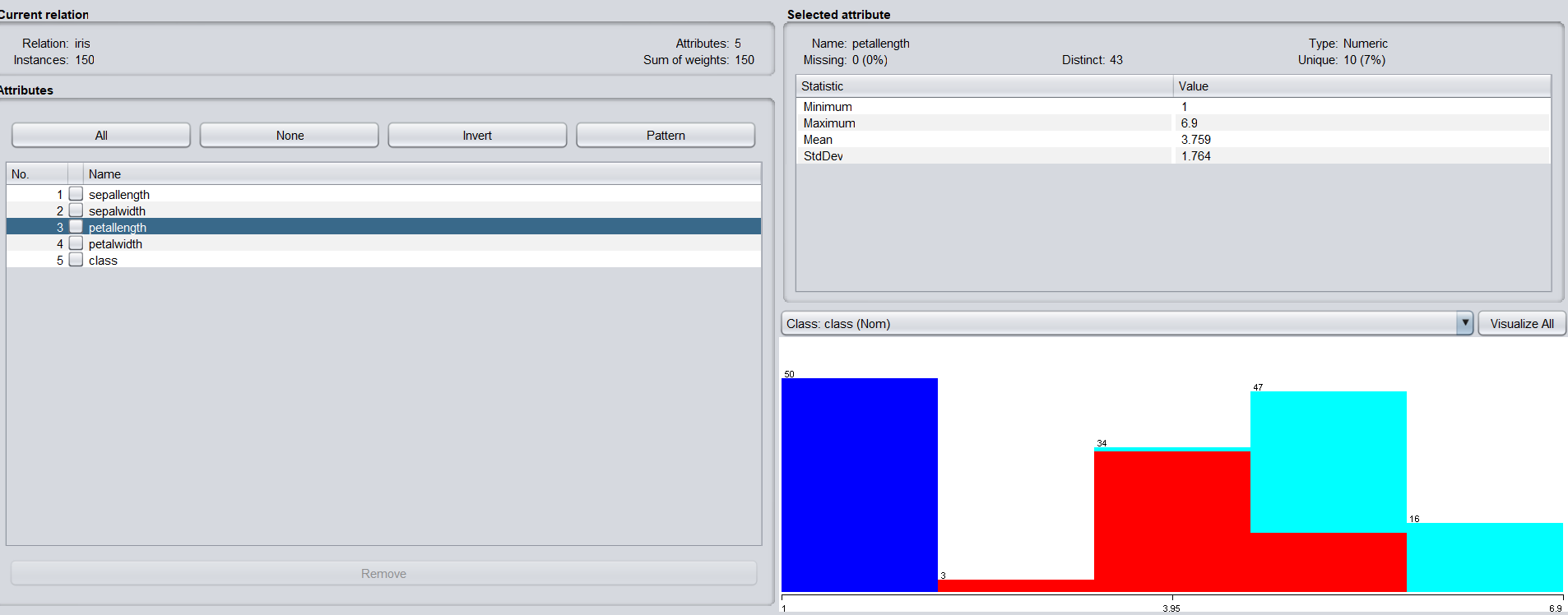
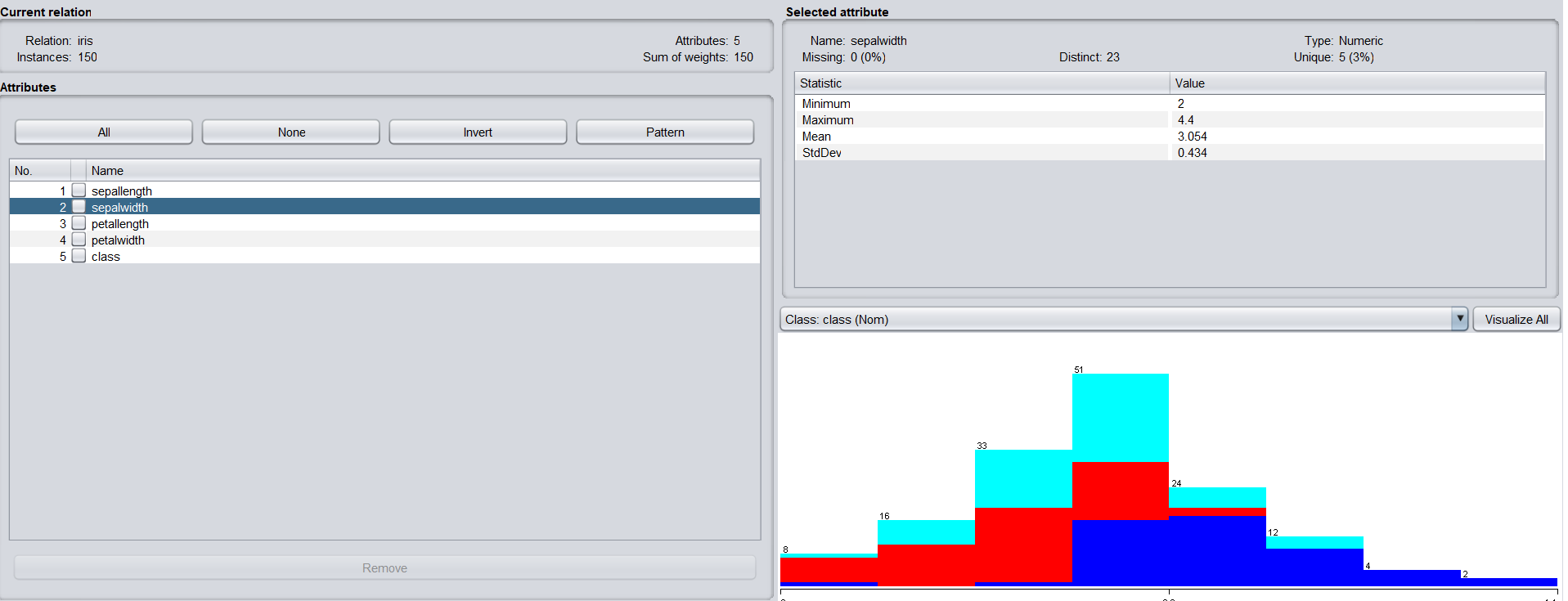
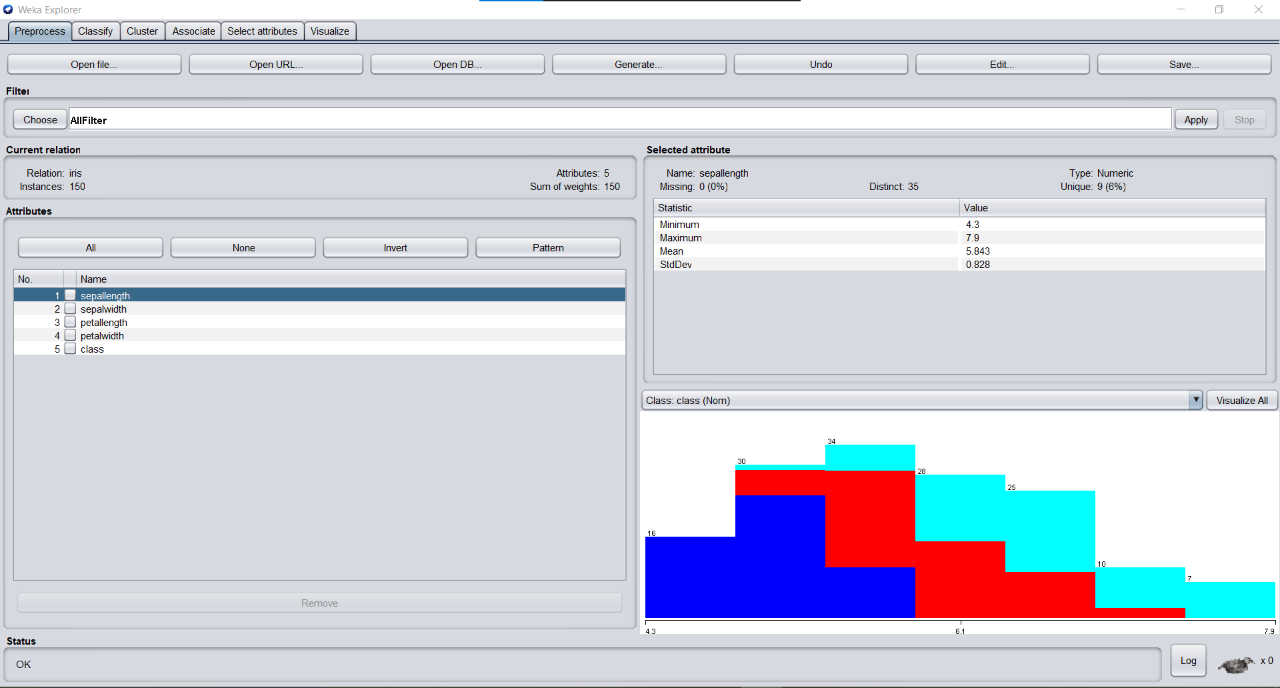


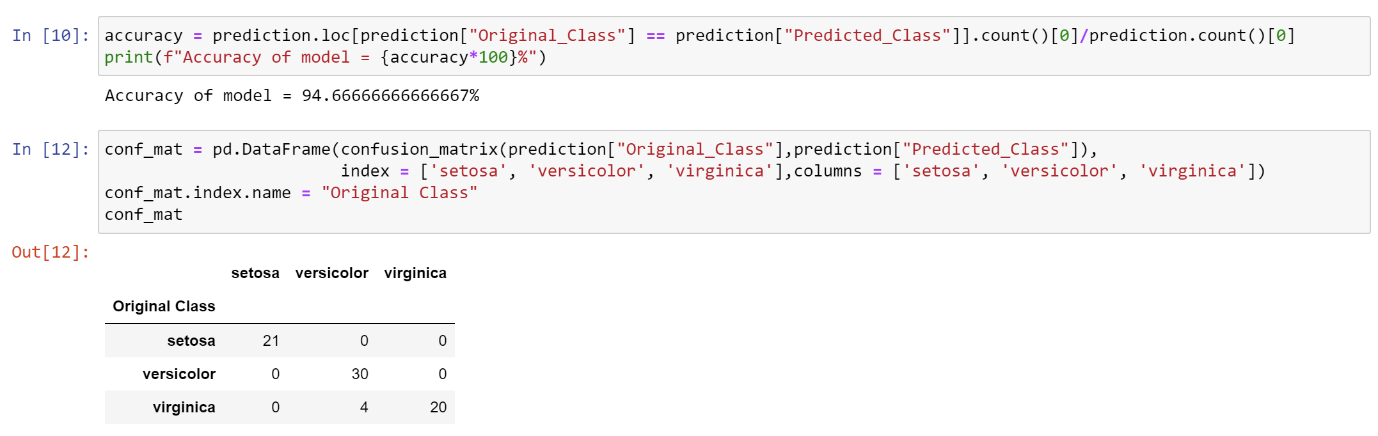
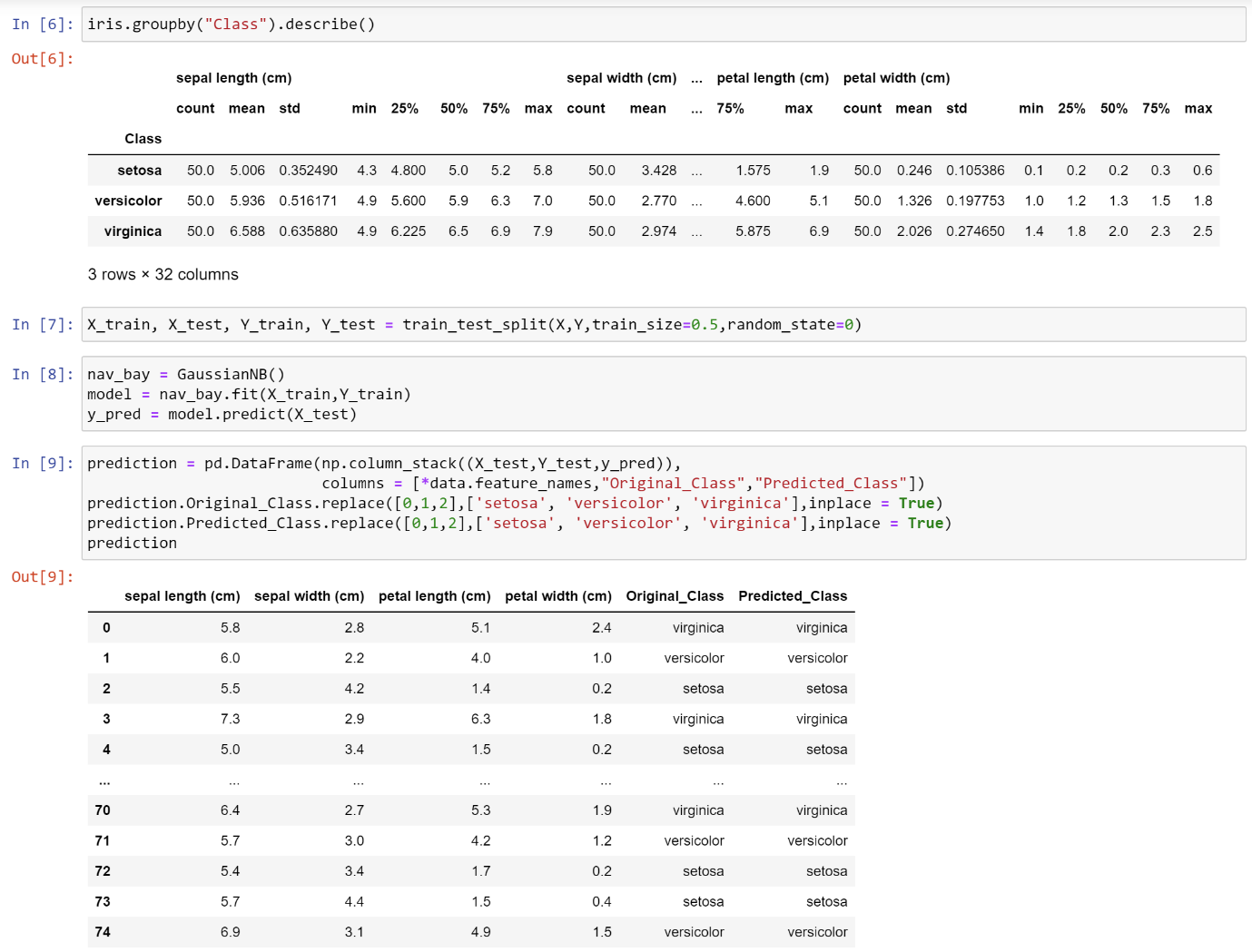
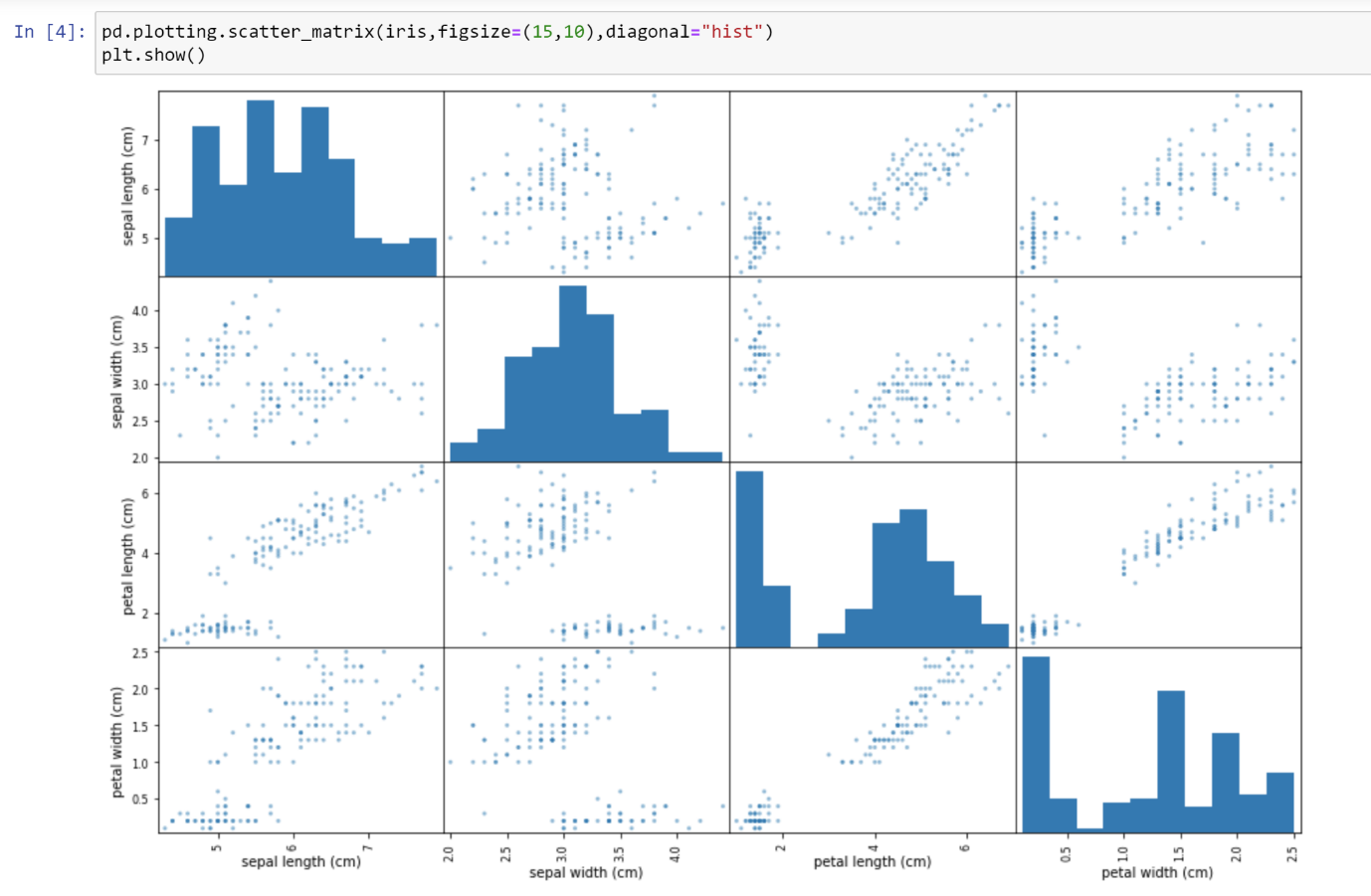
Sometimes assume variance

* is independent of Y (i.e., σi),
* or independent of Xi (i.e., σk)
* or both (i.e., σ)

Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution.

Weka tool output:



Python output:

Conclusion: We have successfully implemented Bayes Algorithm on iris data set using Weka tool and Python libraries.