**DATA WAREHOUSING AND DATA MINING LAB**

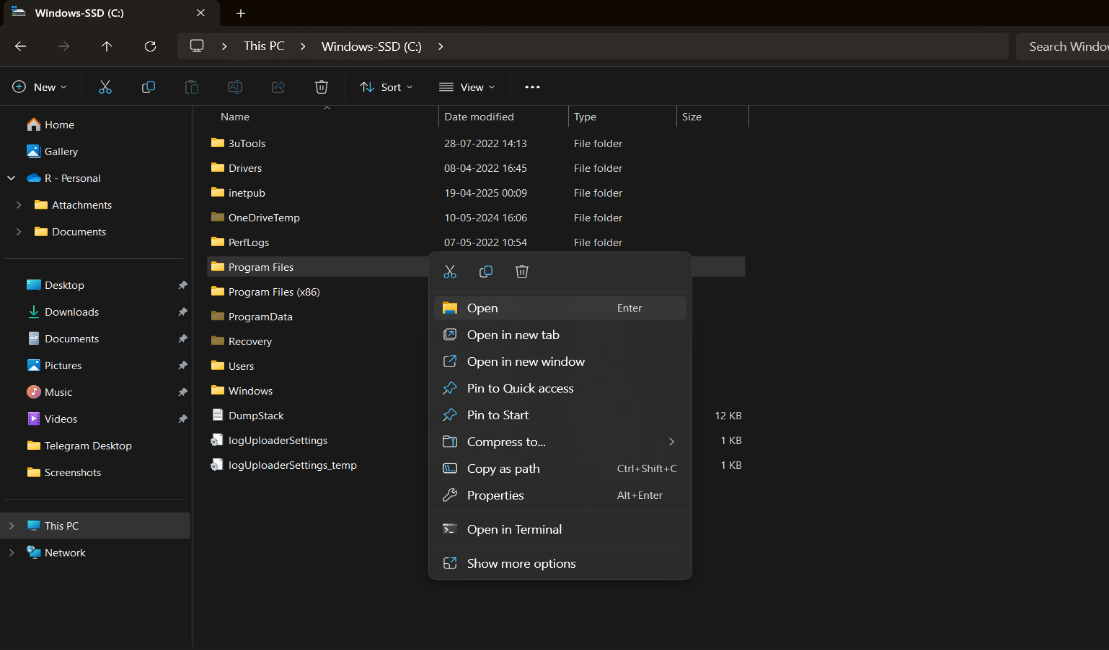
**Experiment 6: Implement Naïve Bayes classifier and evaluate it using accuracy, precision, and recall**

**Aim:**  
 To implement the Naïve Bayes classification algorithm on a given dataset and evaluate its performance using accuracy, precision, and recall metrics.

**Description:**

Naïve Bayes is a simple probabilistic classifier based on **Bayes’ theorem** with a strong assumption of independence between features. Despite this assumption, it performs well in many real-world scenarios such as spam detection and document classification. In this experiment, we train a Naïve Bayes model using WEKA and evaluate its performance.

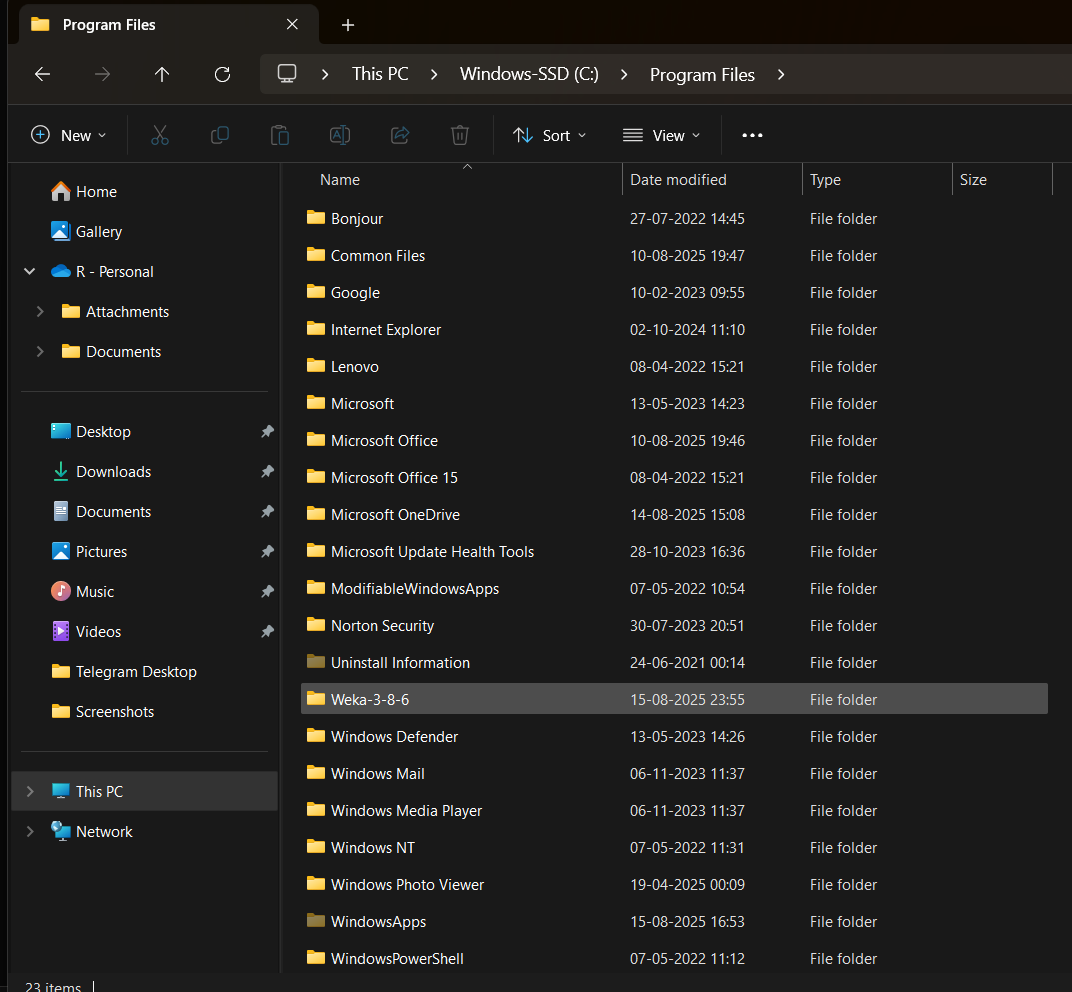
**Naïve Bayes classification using WEKA :  
Step 1:**  
Open **File Explorer**  
Select **This PC** 🡪 Open the **C drive** 🡪 Open **Program Files**

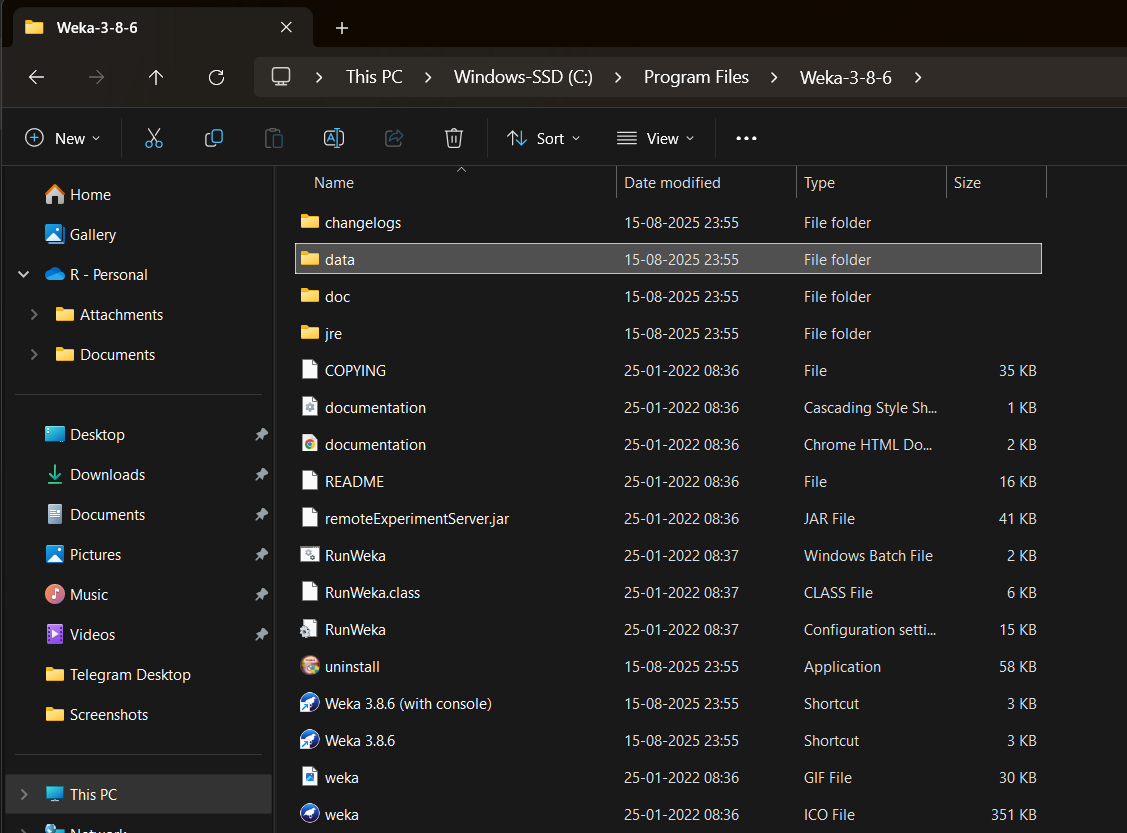
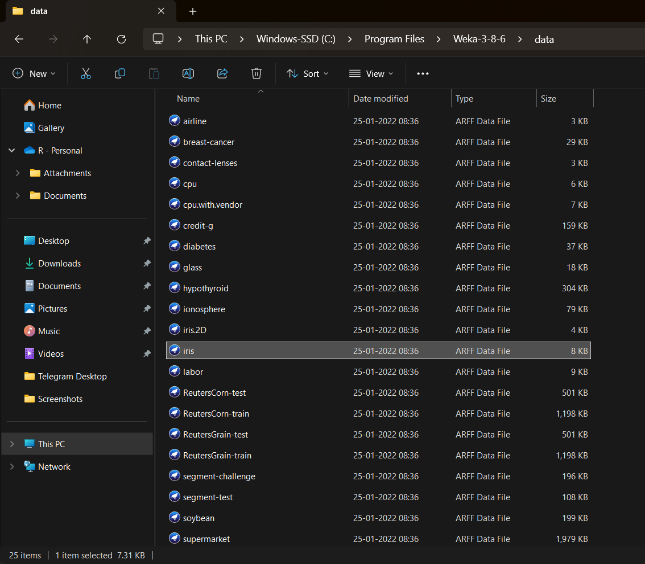


**Step 2 :**Open the **WEKA 3.8.6** folder 🡪 Open the **Data** folder 🡪 Select the **Iris** dataset 🡪 Double-click on it

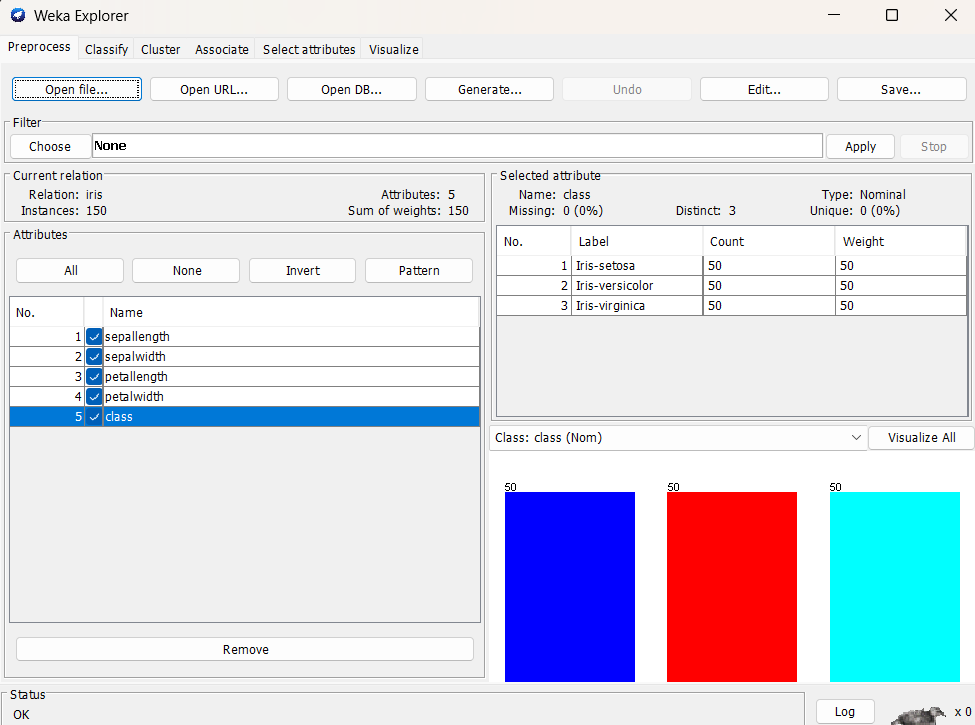
### **Iris Dataset – Information**

* **Introduced by**: Ronald A. Fisher in 1936 (in his paper on discriminant analysis).
* **Purpose**: Used for pattern recognition and classification tasks; one of the most famous benchmark datasets in machine learning.
* **Instances**: 150 flower samples.
* **Attributes**: 4 numerical features (in centimeters):
  1. Sepal length
  2. Sepal width
  3. Petal length
  4. Petal width
* **Class Labels (Species)**: 3 classes, each with **50 samples**:
  1. Iris-setosa
  2. Iris-versicolor
  3. Iris-virginica



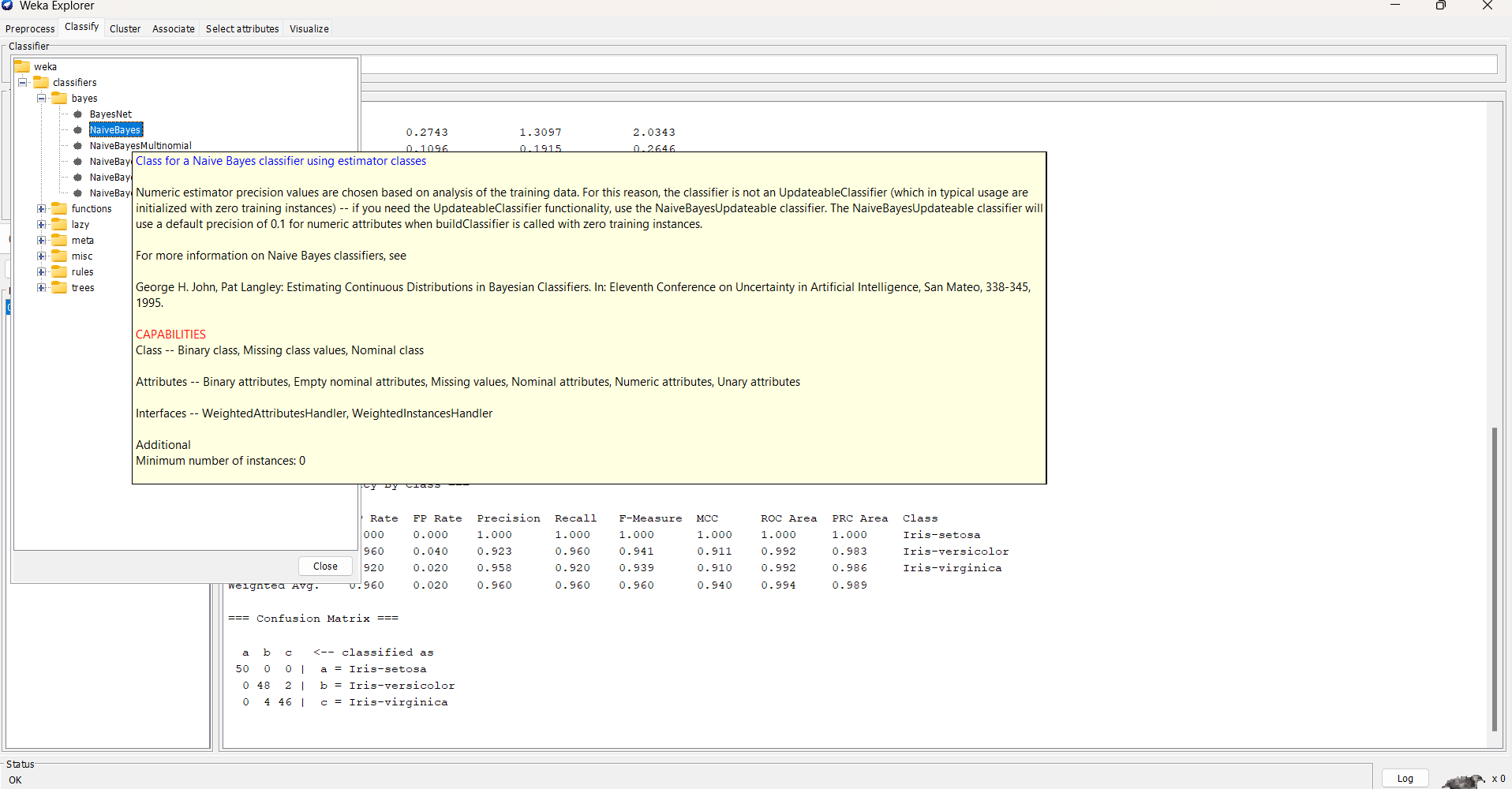
 

**Step 3:**  
 After WEKA opens, select all the parameters displayed there, such as **Sepallength, Sepalwidth, Petallength, Petalwidth,** and **Class**

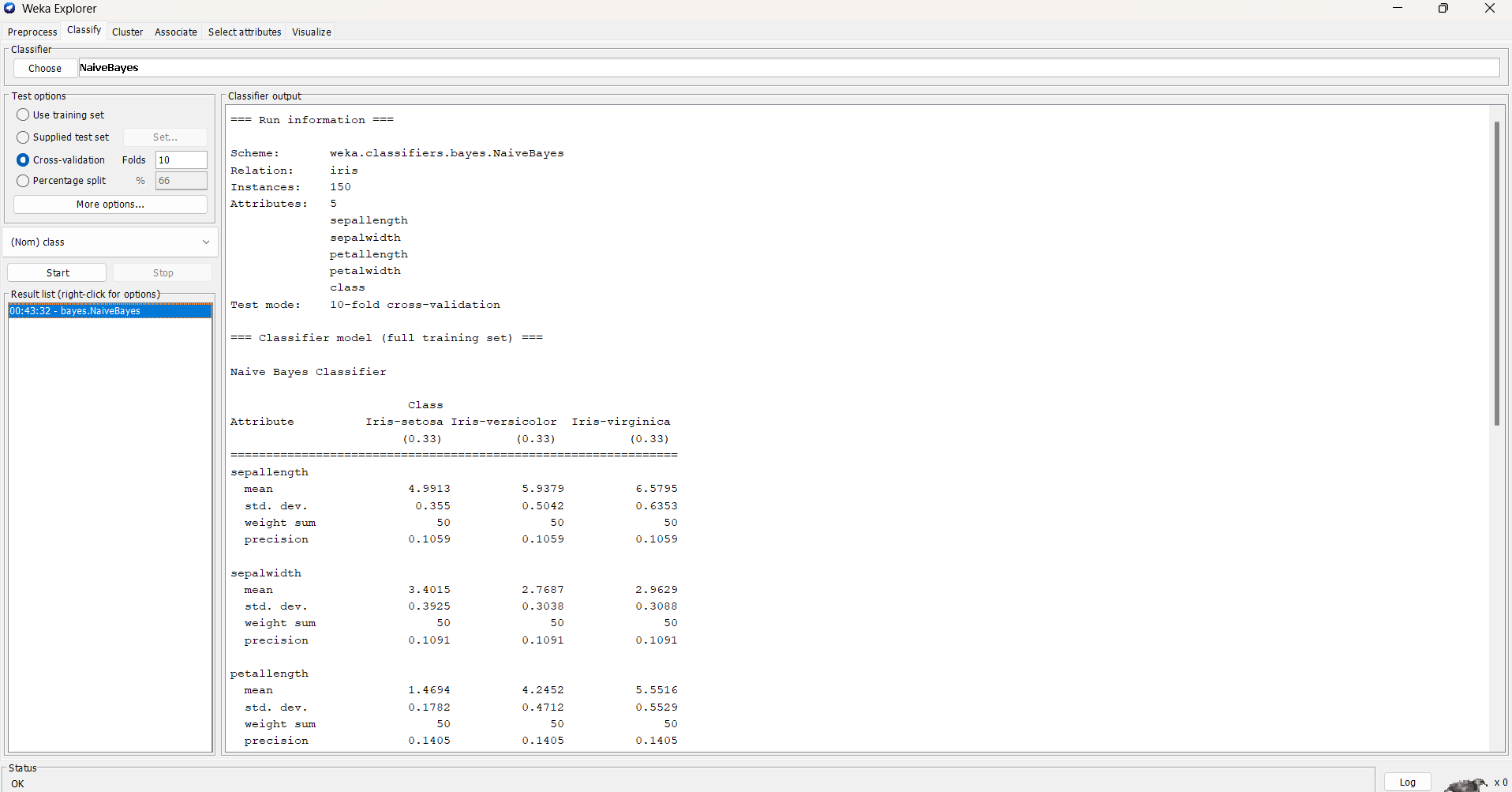


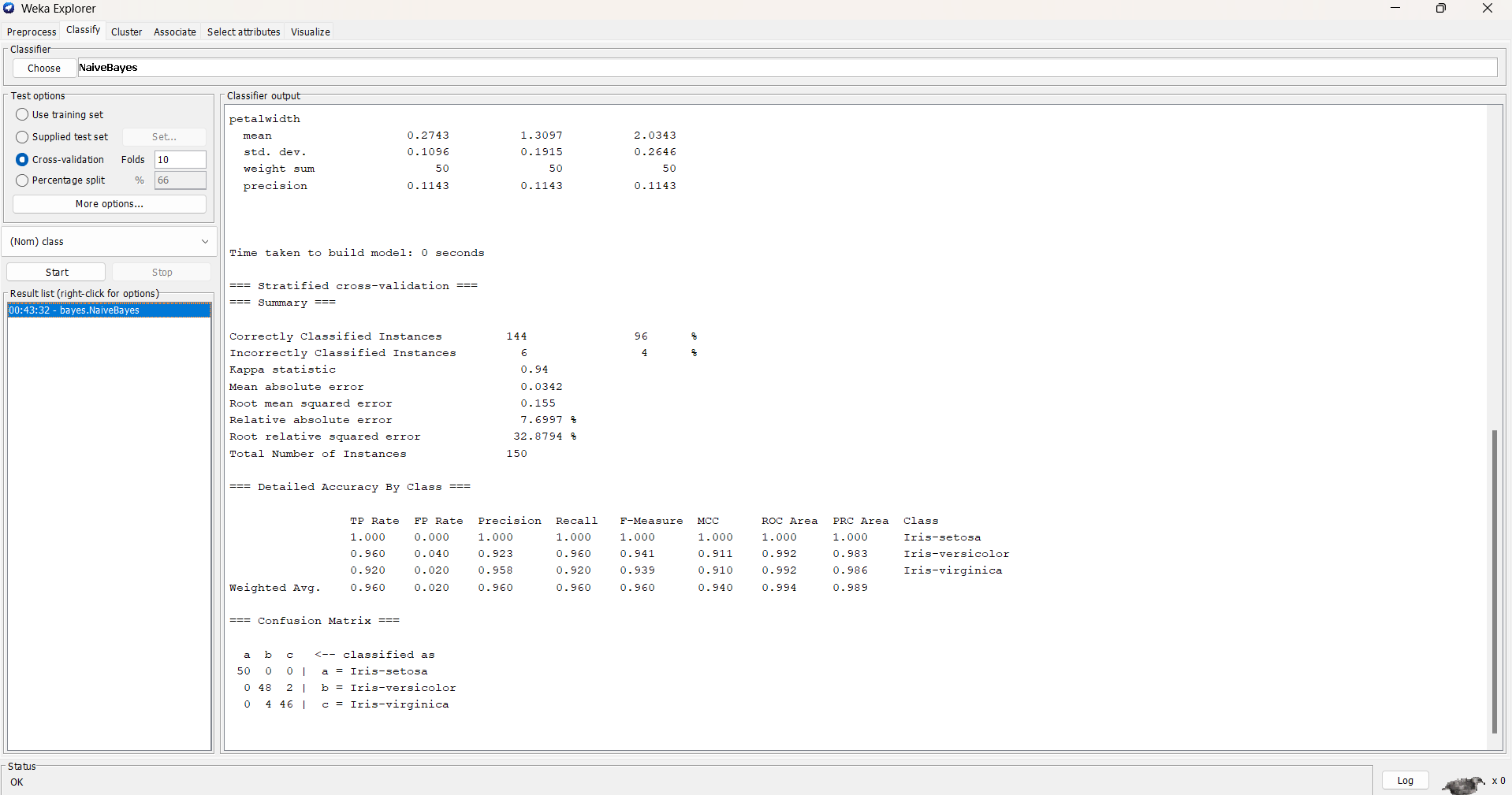
**Step 4:**

Select **Classify** 🡪 Click the **Choose** option 🡪 In the **Classifiers** section, select **Bayes** 🡪 Then, select **Naive Bayes** from the options



**Step 5:** Click on **Start**, and the output will be as follows:





* **Accuracy:** Proportion of total predictions that are correct.
* **Precision (per class):** Of items predicted as that class, how many were correct.
* **Recall (per class):** Of items truly in that class, how many were found.
* **Confusion Matrix:** Table showing predicted vs actual counts by class.

### **Naive Bayes on Iris Dataset – Summary**

* **Dataset**: Iris (150 instances, 5 attributes, 3 classes)
* **Evaluation**: 10-fold cross-validation
* **Accuracy**: **96% (144/150 correct)**
* **Kappa**: 0.94 → strong agreement (Kappa measures how well predictions agree with actual classes beyond chance)
* **Error**: Low (MAE = 0.034, RMSE = 0.1475)

**Class-wise Performance**

* Iris-setosa: 100% correct
* Iris-versicolor: 92% correct (4 misclassified as virginica)
* Iris-virginica: 96% correct (2 misclassified as versicolor)

**Confusion Matrix**

Setosa: 50/50 correct

Versicolor: 46/50 correct

Virginica: 48/50 correct

**Conclusion**: Naive Bayes works very well on Iris (96% accuracy). Setosa is perfectly classified; small confusion occurs between versicolor and virginica.