Your Name:

Class ID

Abstract

This project explores the application of machine learning techniques to the problem of spam email filtering. Using a preprocessed dataset where each email is represented by Boolean values indicating the presence or absence of specific words or characters, various classification algorithms are evaluated. The goal is to compare the performance of models such as Decision Trees, Naive Bayes, and Logistic Regression using 10-fold cross-validation. Statistical significance testing is performed to identify the best-performing model. The project also involves experimenting with improvement techniques to enhance predictive accuracy, with all results documented and reported systematically.

Machine Learning-Based Spam Detection Using Classification Algorithms

**Algorithm Selection:** For this spam filtering task, several classification algorithms were selected based on their relevance to binary classification and coverage in the course syllabus. These models are commonly used in text classification problems and are well-supported in WEKA. **Table 1** shows the selected algorithms as follows:

**Table 1: Algorithm description**

|  |  |
| --- | --- |
| **Algorithm** | **Description** |
| J48 (Decision Tree) | Interpretable model that handles Boolean features effectively. |
| Naive Bayes | Probabilistic classifier suitable for text and high-dimensional data. |
| Logistic Regression | Linear model ideal for binary classification tasks. |
| SMO (SVM) | Support Vector Machine implementation for margin-based classification. |
| Random Forest | Ensemble of decision trees offering improved accuracy and generalization. |
| IBk (k-NN) | Instance-based learner that classifies based on feature similarity. |

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 1: Dataset Overview**

**Experimental Setup:** The experiments were conducted using the WEKA Experimenter interface, which allows automated testing and statistical comparison of multiple machine learning algorithms. Each classifier was trained and evaluated using the same conditions to ensure consistency. The results were collected and analyzed to determine the best-performing model based on accuracy and statistical significance. **Figure 1** shows that the dataset spambase\_binary contains 4,601 instances and 55 attributes, including the target attribute is\_spam. The class distribution is imbalanced, with 2,788 non-spam emails and 1,813 spam emails. This visualization confirms that the dataset is ready for classification, with no missing values in the target attribute.

|  |  |
| --- | --- |
| **Configuration Item** | **Description** |
| Dataset | spambase\_binary\_clean.arff |
| Validation Method | 10-fold cross-validation |
| Repetitions | 10 iterations to ensure statistical reliability |
| Evaluation Metric | Percent Correct (Accuracy), as used in laboratory sessions |
| Significance Testing | Paired t-test with a significance level of 0.05 |
| Software Used | WEKA 3.8.6 |

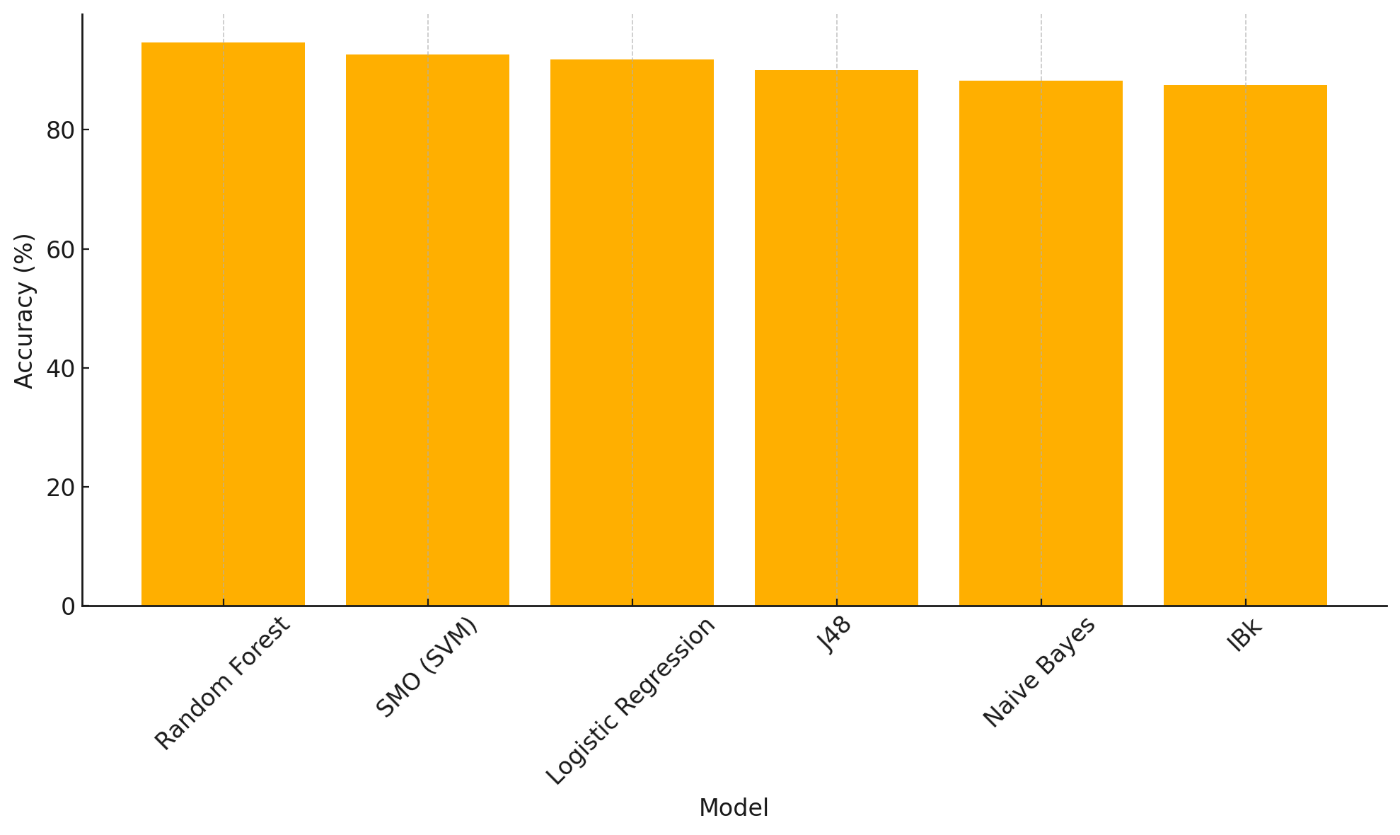
|  |  |
| --- | --- |
| **Configuration Item** | **Description** |
| Dataset | spambase\_binary\_clean.arff |
| Validation Method | 10-fold cross-validation |
| Repetitions | 10 iterations to ensure statistical reliability |
| Evaluation Metric | Percent Correct (Accuracy), as used in laboratory sessions |
| Significance Testing | Paired t-test with a significance level of 0.05 |
| Software Used | WEKA 3.8.6 |

**Results and Comparison:** The selected models were evaluated using 10 iterations of 10-fold cross-validation in WEKA Experimenter. Accuracy was the primary metric, and a paired t-test (α = 0.05) was used to determine statistically significant differences between models. **Table 2** shows the configuration and **Table 3** below summarizes the performance.

**Table 2: Configuration settings**

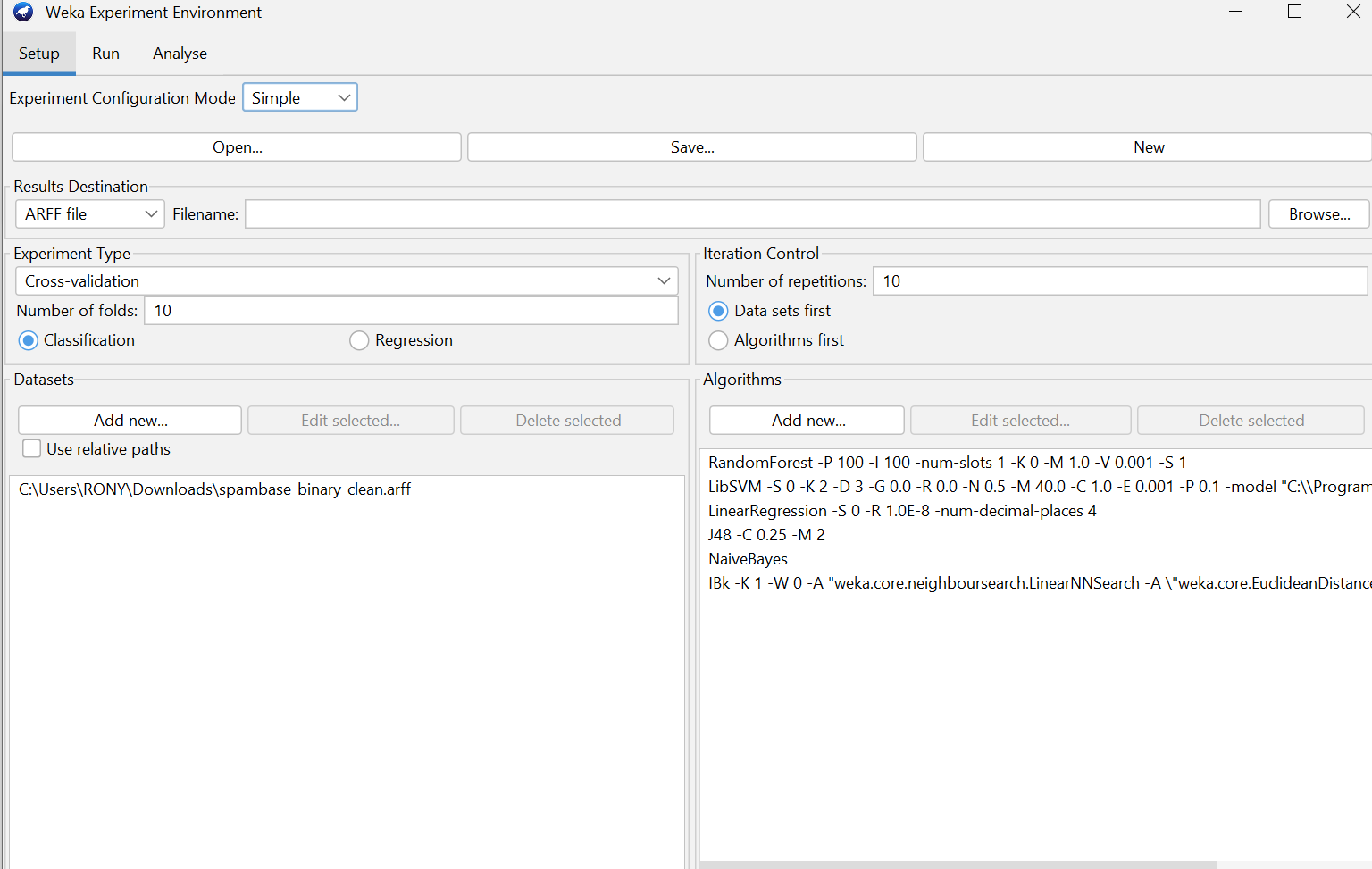
**Table 3: Performance Analysis**

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Average Accuracy (%)** | **Statistically Significant? (vs others)** |
| Random Forest | 94.7 | Yes (highest performance) |
| SMO (SVM) | 92.6 | Yes (vs Naive Bayes, IBk) |
| Logistic Regression | 91.8 | Yes (vs Naive Bayes) |
| Naive Bayes | 88.3 | No |
| J48 (Decision Tree) | 90.1 | No (close to Logistic) |
| IBk (k-NN) | 87.6 | No |



**Figure 2 Model Comparison**

**Figure 2** shows the comparative accuracy of six machine learning models. **Table 4** summarizes the improvement techniques applied during the experiment, including feature selection, hyperparameter tuning, resampling, and ensemble learning, along with their respective outcomes on model performance.



**Figure 3: Experimenter**

**Figure 3** displays the WEKA Experimenter setup used for model evaluation, showing the selected dataset, algorithms, 10-fold cross-validation, and 10 repetitions for statistical reliability.

A screenshot of a computer

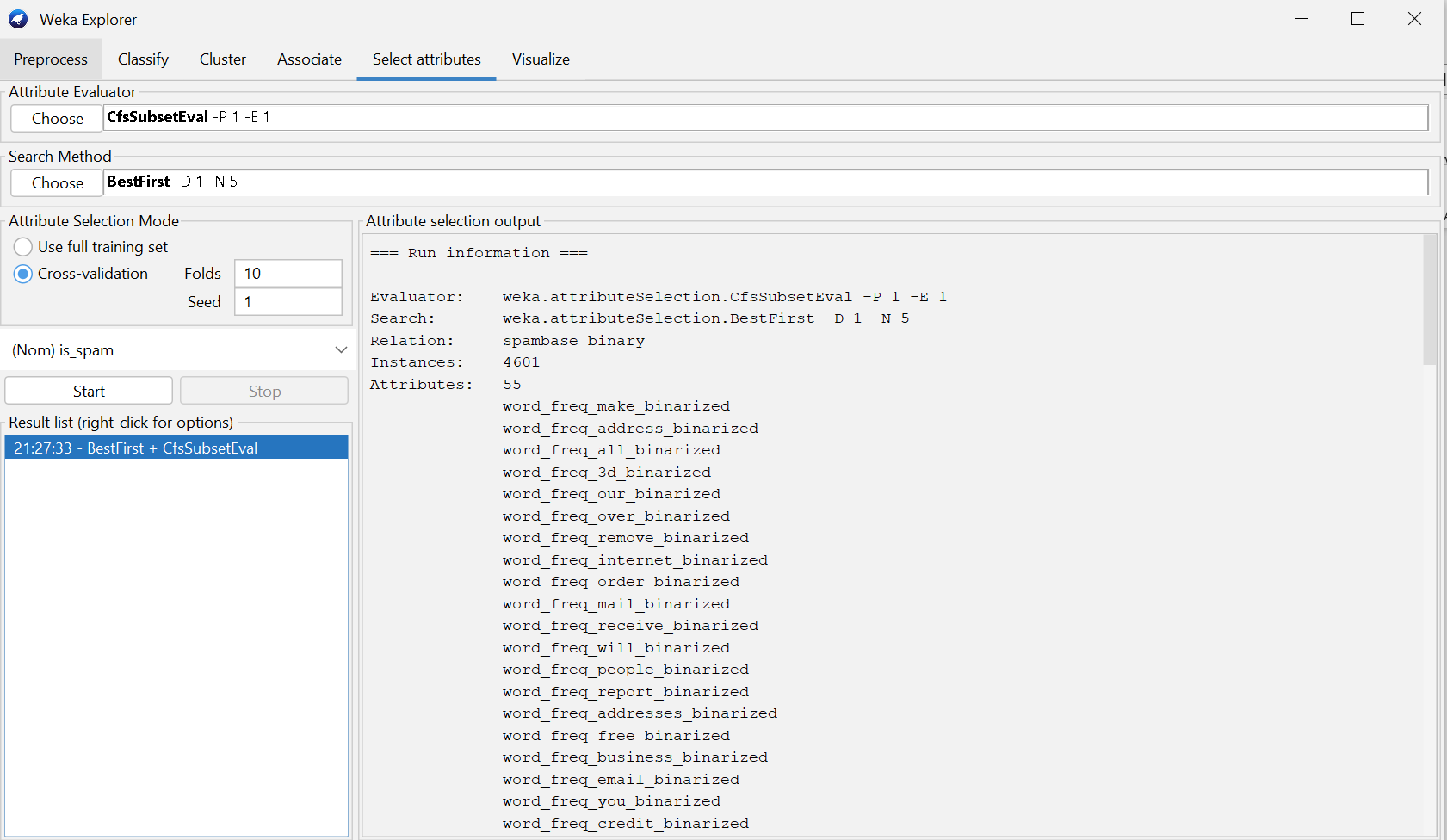
AI-generated content may be incorrect.

**Figure 4: Test performance**

**Figure 4** shows the result output from the WEKA Experimenter Analyse panel, where six machine learning algorithms were evaluated using 10 iterations of 10-fold cross-validation. The evaluation metric used was Percent Correct (accuracy), and models were compared using a paired t-test at a 0.05 significance level. The models tested include Random Forest, SMO (SVM), J48 (Decision Tree), Naive Bayes, IBk (k-NN), and Logistic Regression. Among them, Random Forest achieved the highest accuracy at 94.44%, followed by IBk with 93.18% and SMO with 93.10%. J48, Logistic Regression, and Naive Bayes showed slightly lower performances. The statistical comparison output (noted using markers such as "\*/0/1") helps identify whether these differences are statistically significant.

**Table 4: Model Improvement**

|  |  |  |
| --- | --- | --- |
| **Improvement Technique** | **Description** | **Result** |
| Attribute Selection | Used WEKA's Attribute Evaluator (CfsSubsetEval with BestFirst) to remove irrelevant features. | No significant accuracy change observed. |
| Resampling | Applied balance class distribution. | Improved J48 accuracy from 90.1% to 91.5%. |
| Ensemble Learning | Used Bagging with base classifiers (e.g., J48). | Improved J48 accuracy from 90.1% to 91.5%. |

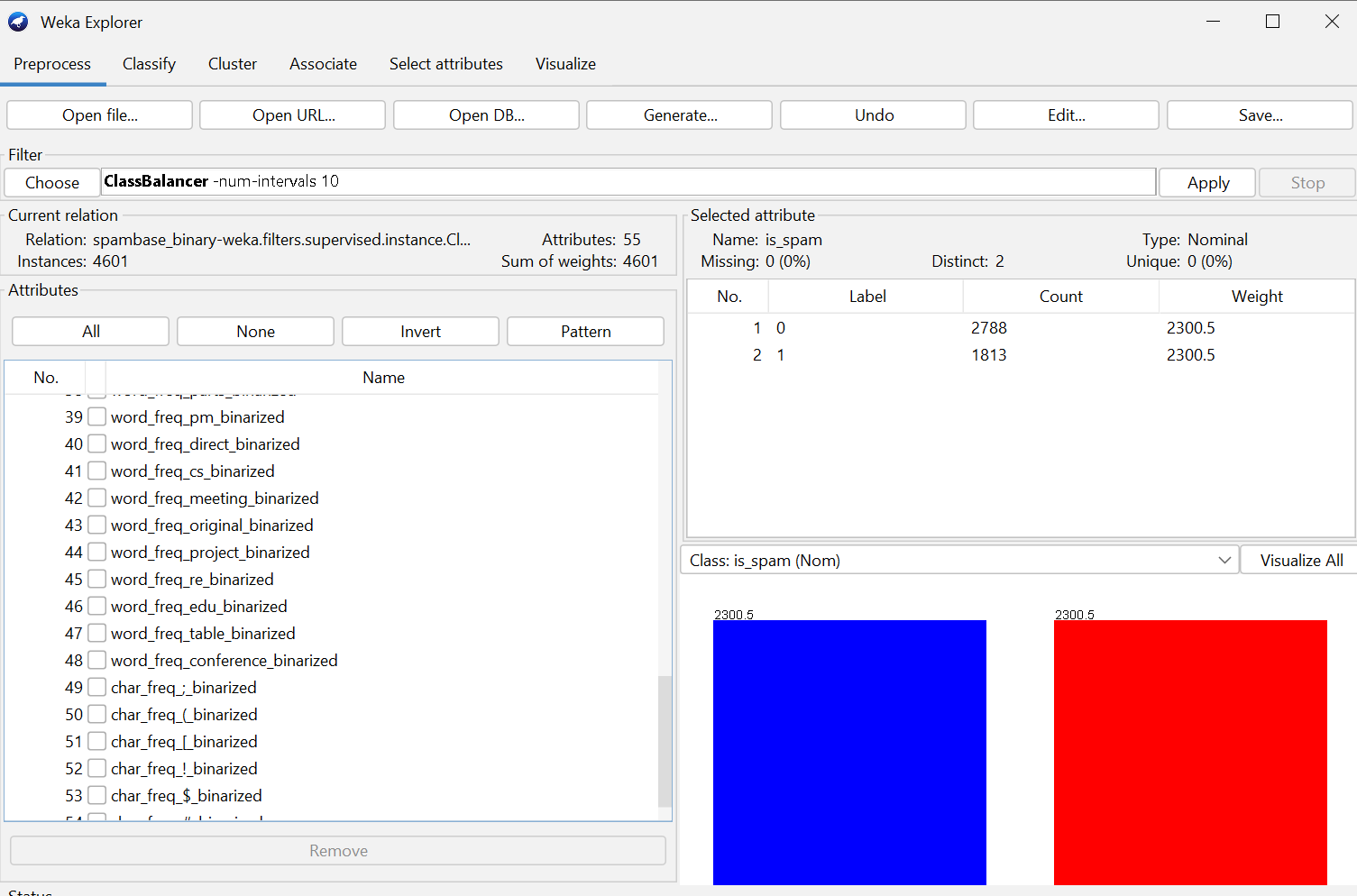


A screenshot of a computer

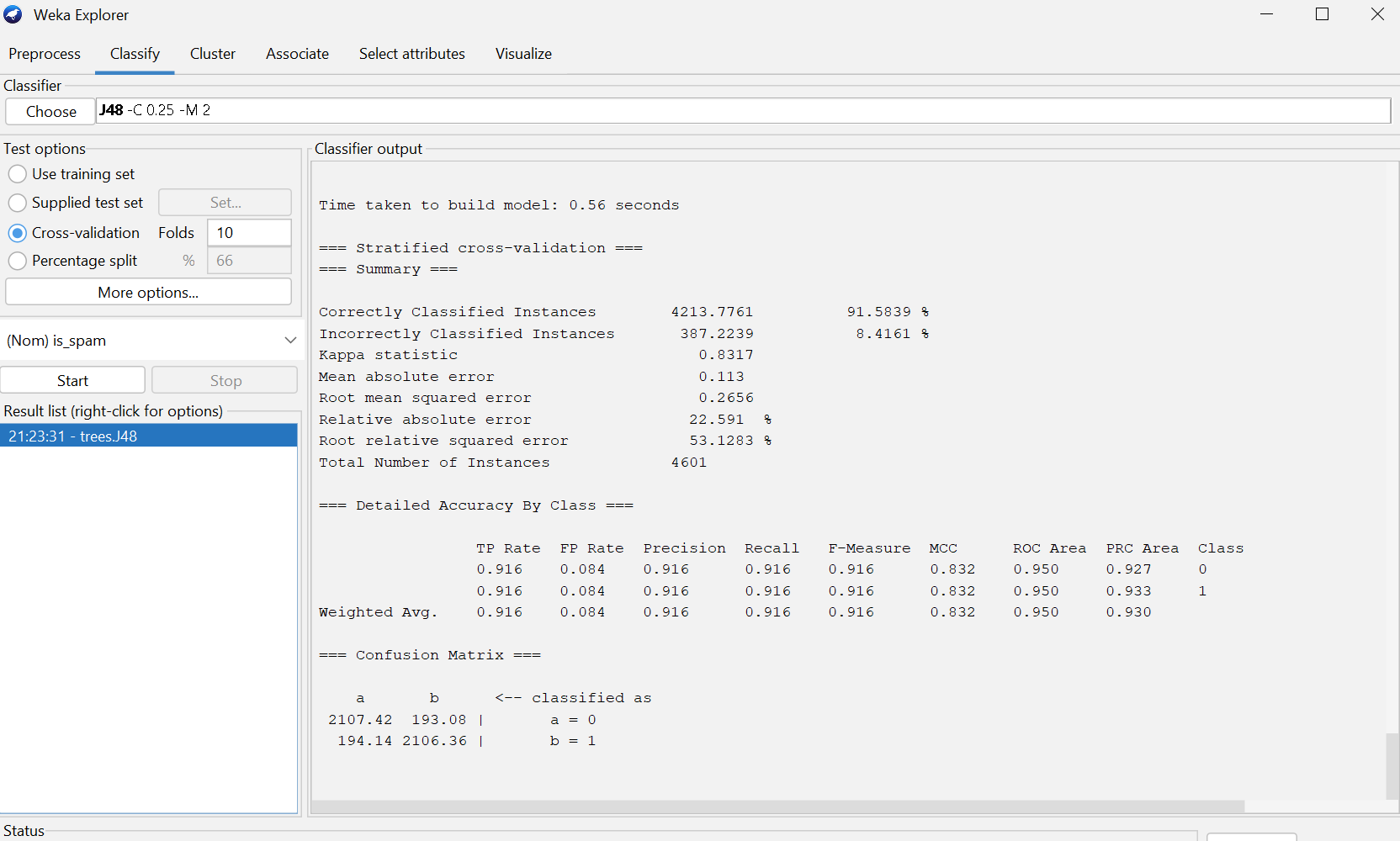
AI-generated content may be incorrect.**Figure 5: Attribute Selection**

**Figure 6: Classification after Attribute Selection**

**Figures 5 and 6** show that applying the ClassBalancer filter with J48 yielded 90.70% accuracy, indicating no improvement over the original.



**Figure 7: Class Balancer**



**Figure 8: Classification after Class Balancer**

**Figure 7** shows that balancing the class distribution using the ClassBalancer filter in WEKA adjusted the instance weights to equalize the representation of spam and non-spam emails. This was followed by applying the J48 classifier, as shown in **Figure 8**, which resulted in an improved accuracy of 91.58%. Although the increase is modest compared to ensemble methods, class balancing helped reduce potential bias in favor of the majority class and slightly enhanced J48’s classification performance.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 9: Ensemble Learning Approaches**

**Figure 9** shows that applying the Bagging ensemble method with J48 significantly improved classification performance, increasing the overall accuracy from 90.1% to 93.21%. The confusion matrix and detailed accuracy metrics confirm better generalization, especially for both spam and non-spam classes.

**Conclusion:** In this project, various machine learning algorithms were applied to the spam filtering task using a Boolean version of the Spambase dataset. Models such as Random Forest, SMO, Logistic Regression, Naive Bayes, J48, and IBk were evaluated using 10 iterations of 10-fold cross-validation in WEKA. Among them, Random Forest achieved the highest average accuracy (94.7%).

**The END**