Using Weka for Data Analysis

Raghda Al taei

2024

1 Introduction

Weka is a powerful open-source software suite for machine learning and data mining tasks. It provides a collection of algorithms and tools for classification, regression, clustering, and visualization. In this document, we describe the steps taken to analyze a dataset using decision trees and random forests in Weka, highlighting how we generated results and insights from the data.

2 Step 1: Loading the Data

We started by loading the dataset using the Weka interface. The dataset used was the *labor* dataset, which can be found within the sample datasets included with Weka.

1. Open Weka and select the **Explorer** option.
2. Navigate to the **Preprocess** tab.
3. Click on the **Open file** button and select the *labor.arff* file from the Weka sample datasets.

3 Step 2: Building Decision Trees

To build a decision tree, we utilized the J48 algorithm, Weka's implementation of the C4.5 algorithm. The steps included:

1. Navigate to the **Classify** tab.
2. Choose the **J48** classifier from the **Classifier** dropdown menu.
3. Set the following parameters:
    - **Validation**: We opted for **Cross-validation** with 10 folds to ensure reliable accuracy estimates.
4. Click on the **Start** button to build the model.

3.1 Results from Decision Tree

Upon completion of the model building, Weka provided us with the following results:

- **Accuracy**: The model classified 73.68% of instances correctly.
- **Confusion Matrix**: A confusion matrix was generated, showing the counts of true positives, false positives, true negatives, and false negatives.

4 Step 3: Pruning the Decision Tree

To improve the decision tree's performance and avoid overfitting, we enabled pruning in the J48 algorithm. The steps were similar to the previous model, with the key difference being:

1. In the **J48** configuration, we set the **unpruned** parameter to **False** (which is the default setting).

4.1 Results from Pruned Decision Tree

After training the pruned model, we observed an improvement in performance:

- **Accuracy**: The accuracy increased to 78.95%.
- **Kappa Statistic**: This also improved, indicating a better agreement between the predicted and actual classes.

5 Step 4: Building the Random Forest Model

Next, we explored the **Random Forest** algorithm to further enhance our classification results. The process included:

1. Selecting the **RandomForest** classifier from the **Classifier** dropdown menu.
2. Configuring the number of trees in the ensemble and other relevant parameters.
3. Running the model with the same 10-fold cross-validation.

5.1 Results from Random Forest

The Random Forest model yielded the following outcomes:

- **Accuracy**: The model achieved an impressive accuracy of 89.47%.
- **Confusion Matrix**: The confusion matrix showed a significant reduction in classification errors compared to the decision tree models.

6 Step 5: Comparing the Models

We compared the results of the decision tree and random forest models to evaluate their effectiveness:

- **Decision Tree without Pruning**: 73.68% accuracy.
- **Pruned Decision Tree**: 78.95% accuracy.
- **Random Forest**: 89.47% accuracy.

7 Conclusion

The analysis demonstrated that the Random Forest algorithm outperformed both versions of the decision tree in terms of accuracy and reliability. This is attributed to the ensemble nature of Random Forest, which mitigates the overfitting issues often encountered with single decision trees. Overall, Weka proved to be an effective tool for our data analysis tasks, providing a user-friendly interface and powerful machine learning algorithms.