Assignment 1

Prepared by

Hussam Aldossary

Supervised by

Dr. Esam Mohammed Othman

Table of Contents

1. INTRODUCTION 3

1.1. PURPOSE 3

1.2. SCOPE 3

2. DATASET ANALYSIS AND PREPARATION 4

2.1. Rice: Cammeo &Osmancik: Binary Classification 4

2.1.1. Description 4

2.1.2. features 4

2.1.3. WEKA Analysis 5

2.1.4. Results 13

2.2. Car Evaluation: Multi-class Classification 15

2.2.1. Description 15

2.2.2. features 15

2.2.3. WEKA Analysis 16

2.2.4. Results 29

2.3. Computer Hardware: Regression 31

2.3.1. Description 31

2.3.2. features 31

2.3.3. WEKA Analysis 32

2.3.4. Results 39

3. REFERENCES .....

1 INTRODUCTION

1.1 Purpose

The purpose of this assignment is to apply practical data mining techniques using the WEKA (Waikato Environment for Knowledge Analysis) toolset to solve real-world classification and regression problems. Through hands-on analysis of three distinct datasets sourced from the UCI Machine Learning Repository, this project aims to:

1. Develop proficiency in the **end-to-end data mining process**, from dataset acquisition and preparation to model development and evaluation.
2. **Compare and contrast** the application of data mining methodologies across different problem types: binary classification, multi-class classification, and regression.
3. **Evaluate model performance** using appropriate metrics (e.g., accuracy, precision, recall, RMSE) to draw meaningful insights about the data and the algorithms applied.
4. Demonstrate the utility of the **WEKA platform** as a comprehensive tool for exploratory data analysis, preprocessing, and predictive modeling.

This report documents the methodology, analysis, and findings for each dataset, serving as a structured record of the analytical process and its outcomes.

1.2 SCOPE

The scope of this project is defined by the selection and analysis of three specific datasets and the use of the WEKA software. The boundaries of the work are as follows:

1. **Datasets**: The analysis is confined to three publicly available datasets from the UCI Machine Learning Repository:
    - One dataset for a **binary classification** task.
    - One dataset for a **multi-class classification** task.
    - One dataset for a **regression** (estimation) task.
2. **Tool**: All data preprocessing, exploration, and modeling tasks are performed exclusively within the **WEKA Explorer** environment.
3. **Process**: For each dataset, the project follows a standardized workflow:
    - **Dataset Analysis**: Understanding the domain, attributes, and target variable.
    - **Data Preparation**: Loading and preprocessing data in ARFF format suitable for WEKA.
    - **Model Development & Evaluation**: Applying suitable classification or regression algorithms and evaluating their performance using standard metrics and validation techniques.
4. **Deliverable**: The final output is this comprehensive report, which includes dataset descriptions, methodological steps, analysis of results supported by WEKA screenshots, and a comparative discussion of findings.

Out of Scope:

- Deployment of models in a production environment.
- Extensive feature engineering beyond basic WEKA filters.
- Comparative analysis with other data mining tools (e.g., R, Python's scikit-learn).
- In-depth theoretical derivation of the algorithms used.

2 DATASET ANALYSIS AND PREPARATION

This section provides a detailed overview of the three datasets selected from the UCI Machine Learning Repository for analysis. Each dataset represents a distinct machine learning task: binary classification, multi-class classification, and regression. For each dataset, a description of the problem domain, a summary of its features, and notes on data preparation for WEKA are provided.

2.1 Dataset 1: Binary Classification – Rice (Cammeo and Osmancik)

2.1.1 Description

The Rice dataset addresses a **binary classification** problem in the field of agricultural biology. The goal is to classify images of rice grains into one of two species: **Cammeo** or **Osmancik**. These are prominent rice varieties grown in Turkey. The dataset was created by processing images of 3,810 individual rice grains and extracting seven morphological features based on their geometric properties. The challenge is to use these physical measurements to accurately distinguish between the two species, which is valuable for automated quality control and agricultural sorting systems.

2.1.2 Features

All seven predictive features are **numerical**, derived from image processing. The target variable is a binary class label.

Feature Name	Data Type	Description
Area	Integer	Number of pixels within the boundaries of the rice grain.
Perimeter	Continuous	Distance in pixels around the boundaries (circumference) of the grain.
Major_Axis_Length	Continuous	Length in pixels of the longest axis (line) that can be drawn on the grain.
Minor_Axis_Length	Continuous	Length in pixels of the shortest axis that can be drawn on the grain.

Eccentricity	Continuous	Measures how round the ellipse matching the grain's shape is (0=circle).
Convex_Area	Integer	Pixel count of the smallest convex shape that can contain the grain.
Extent	Continuous	Ratio of the grain's pixel area to the area of its bounding box.
Class (Target)	Binary	Specifies the rice species: Cammeo or Osmancik.

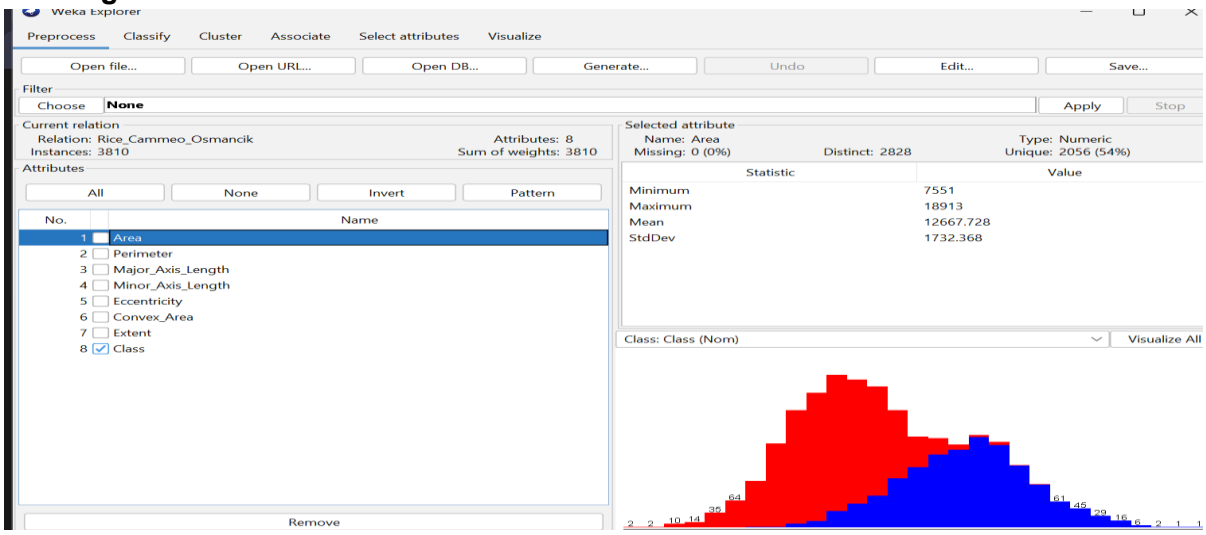
Key Dataset Facts:

- **Number of Instances:** 3,810
- **Number of Features:** 7 (predictive) + 1 (target)
- **Missing Values:** None
- **Source File:** Rice_Cammeo_Osmancik.arff

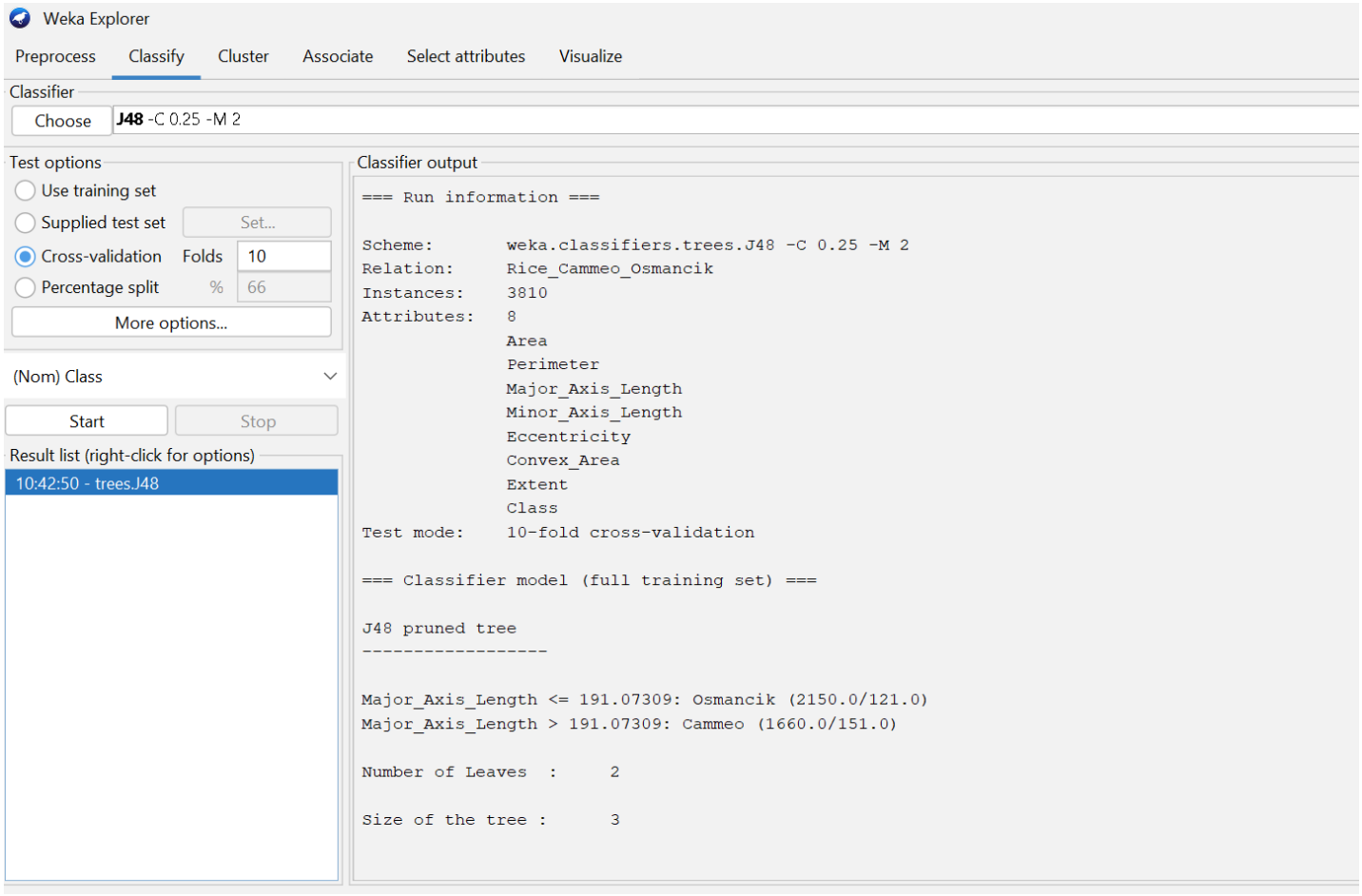
2.1.3 Data Preparation for WEKA

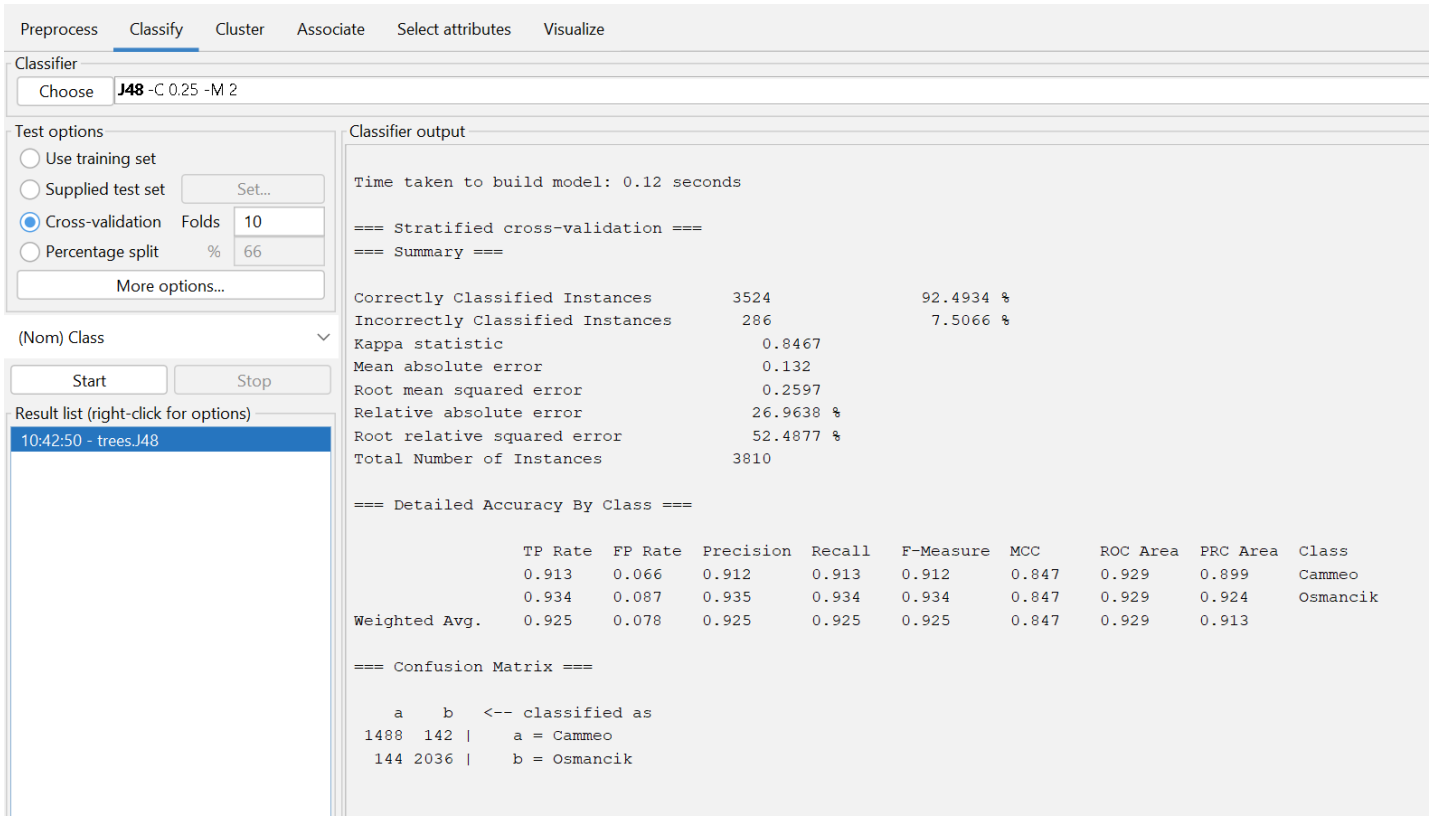
The dataset is readily available in the ARFF (Attribute-Relation File Format) from the UCI repository, which is natively compatible with WEKA. The file was downloaded directly and loaded into the WEKA Explorer without requiring format conversion. No missing value imputation or normalization was performed as a first step, as the data is complete.

Loading data

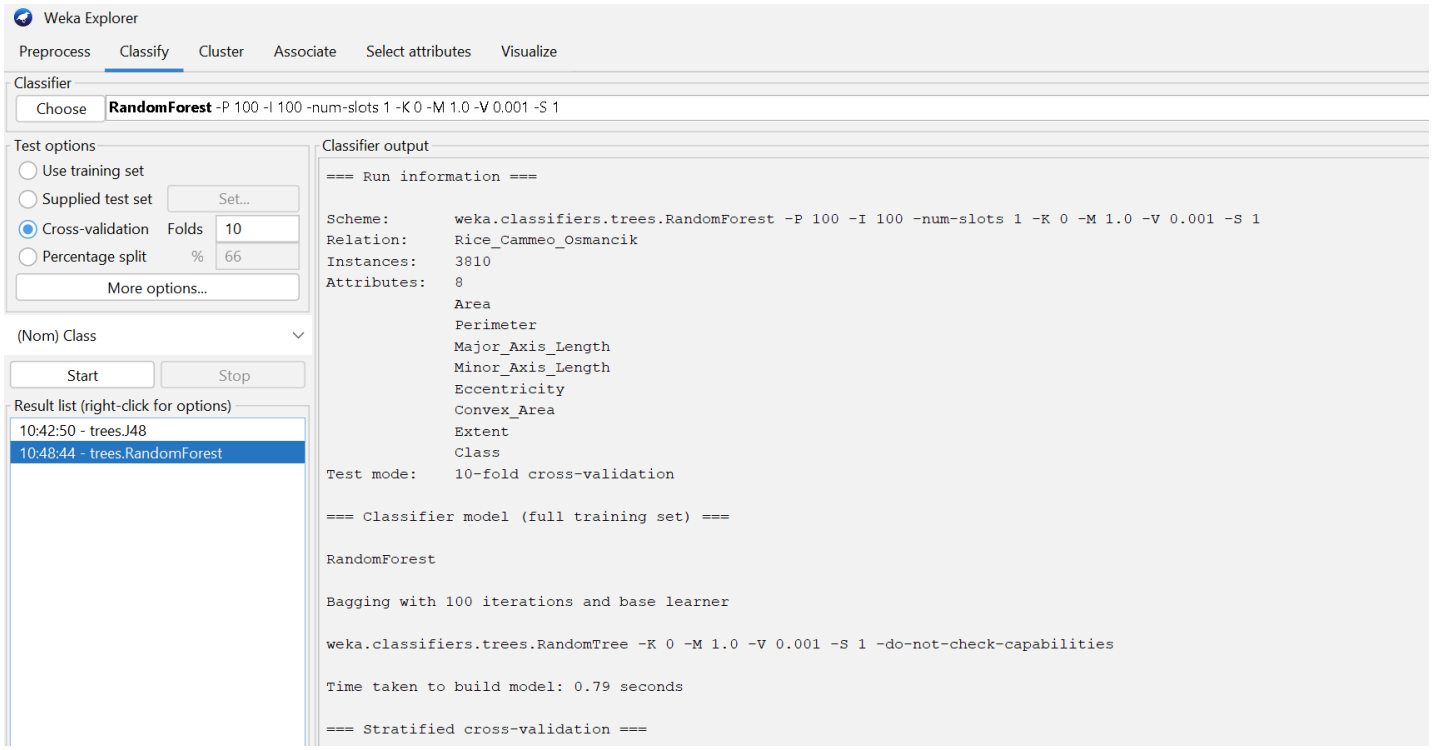


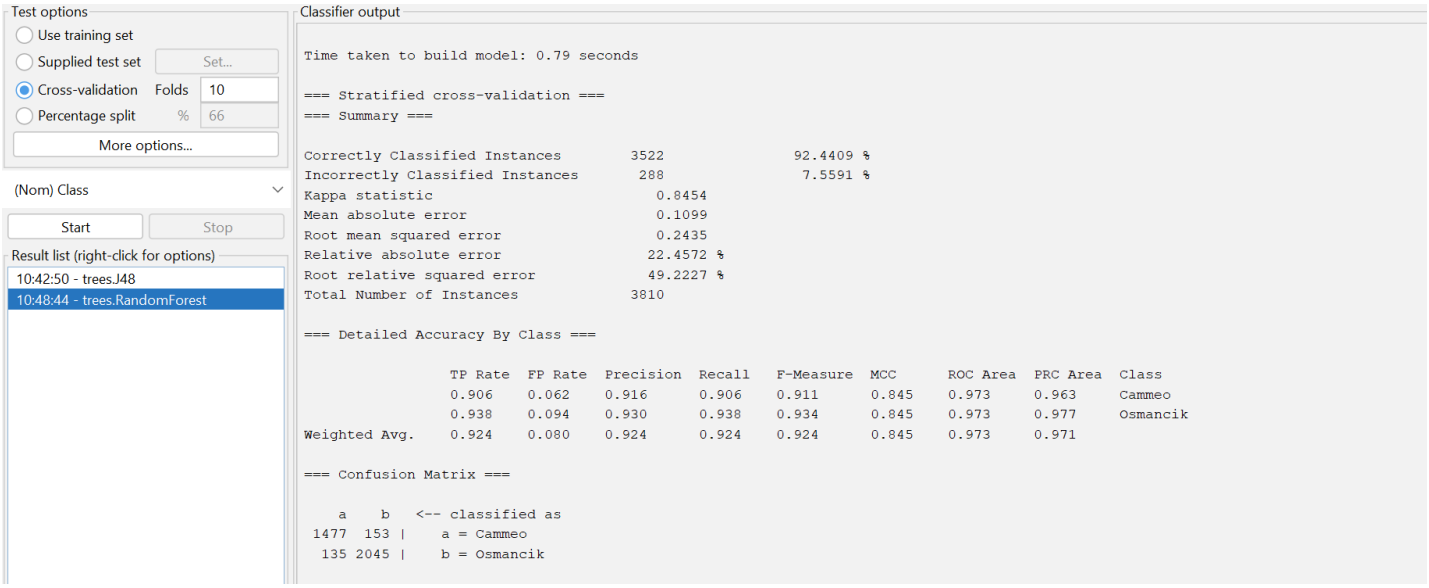
J48 (C4.5 Decision Tree)



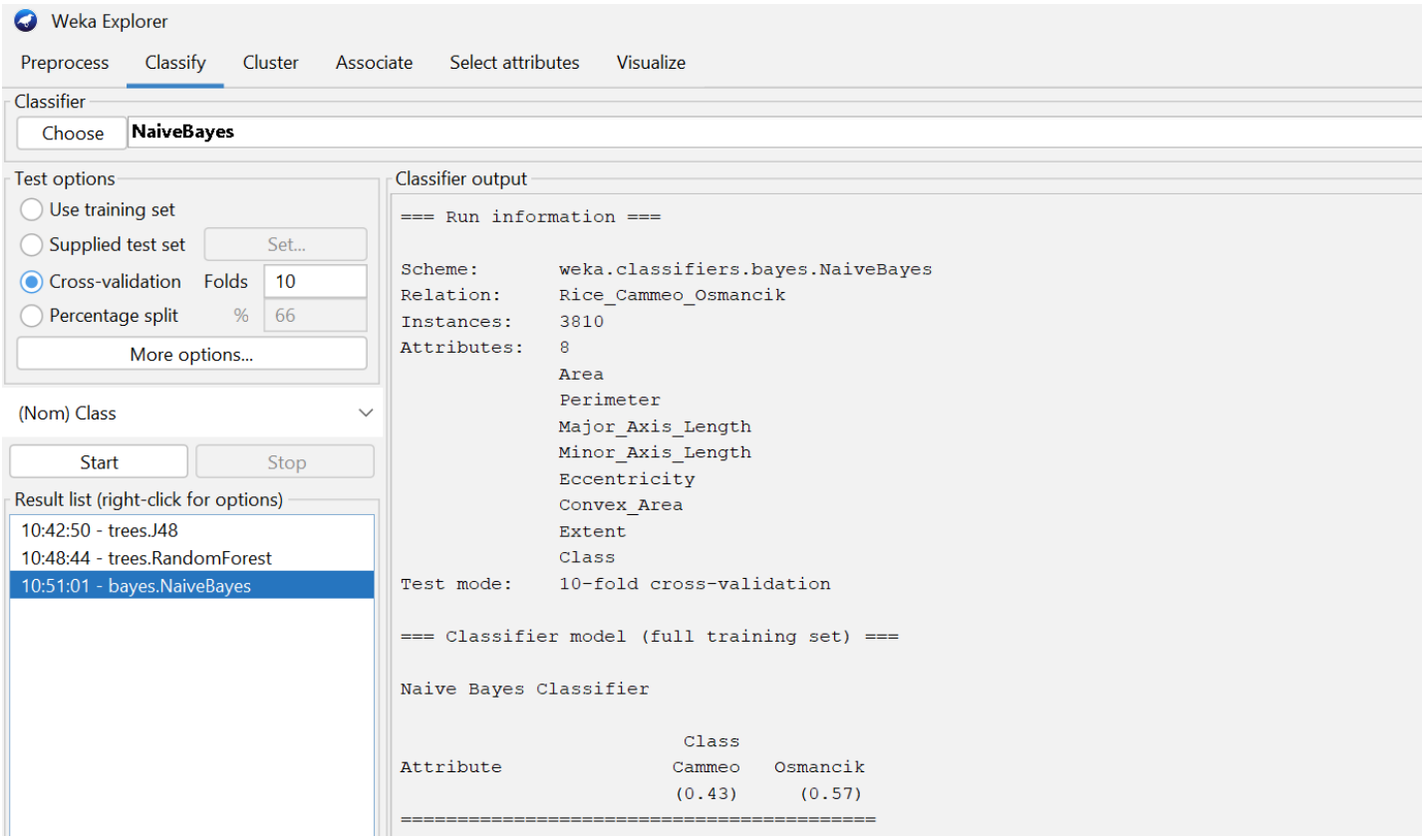


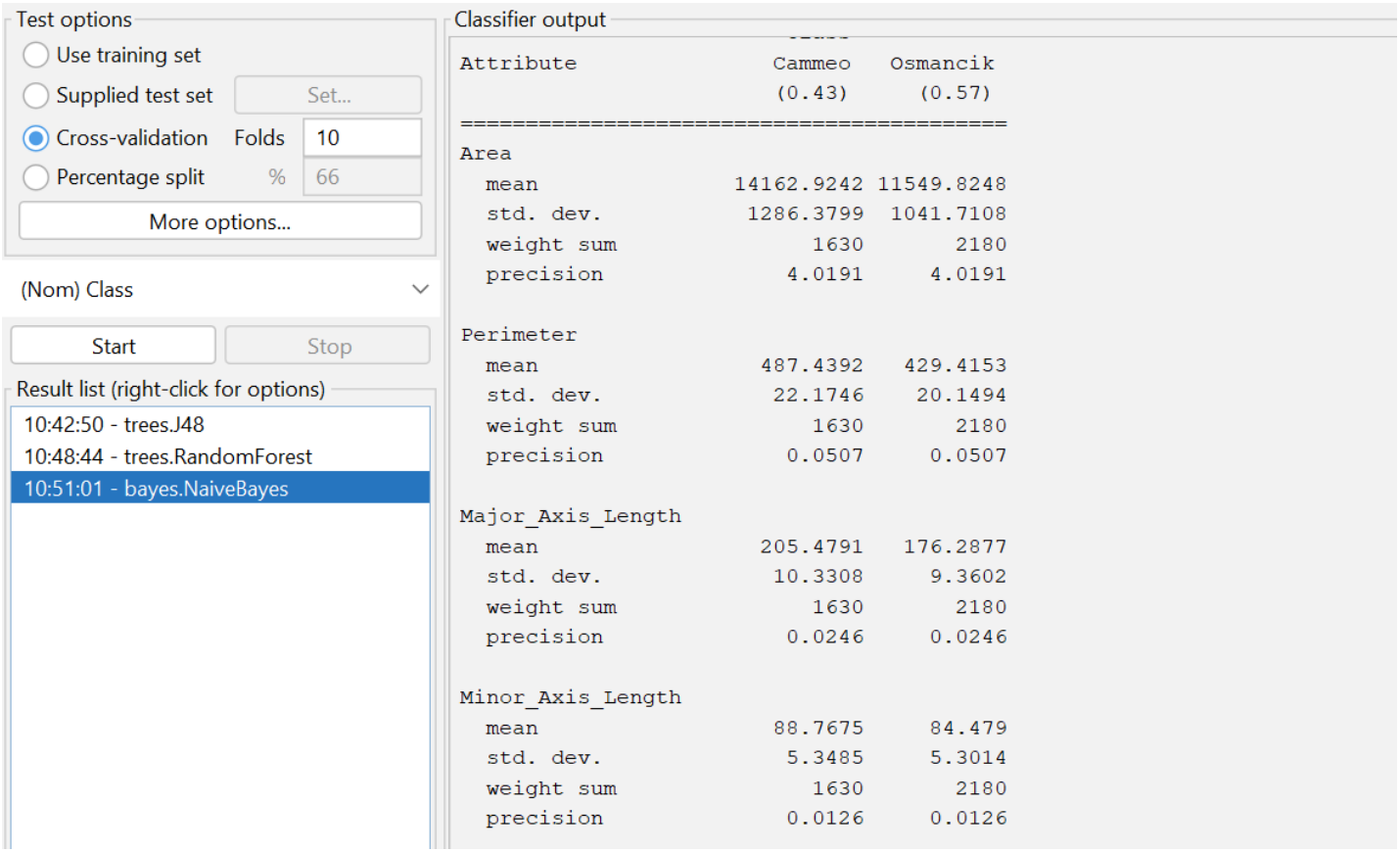
Random Forest

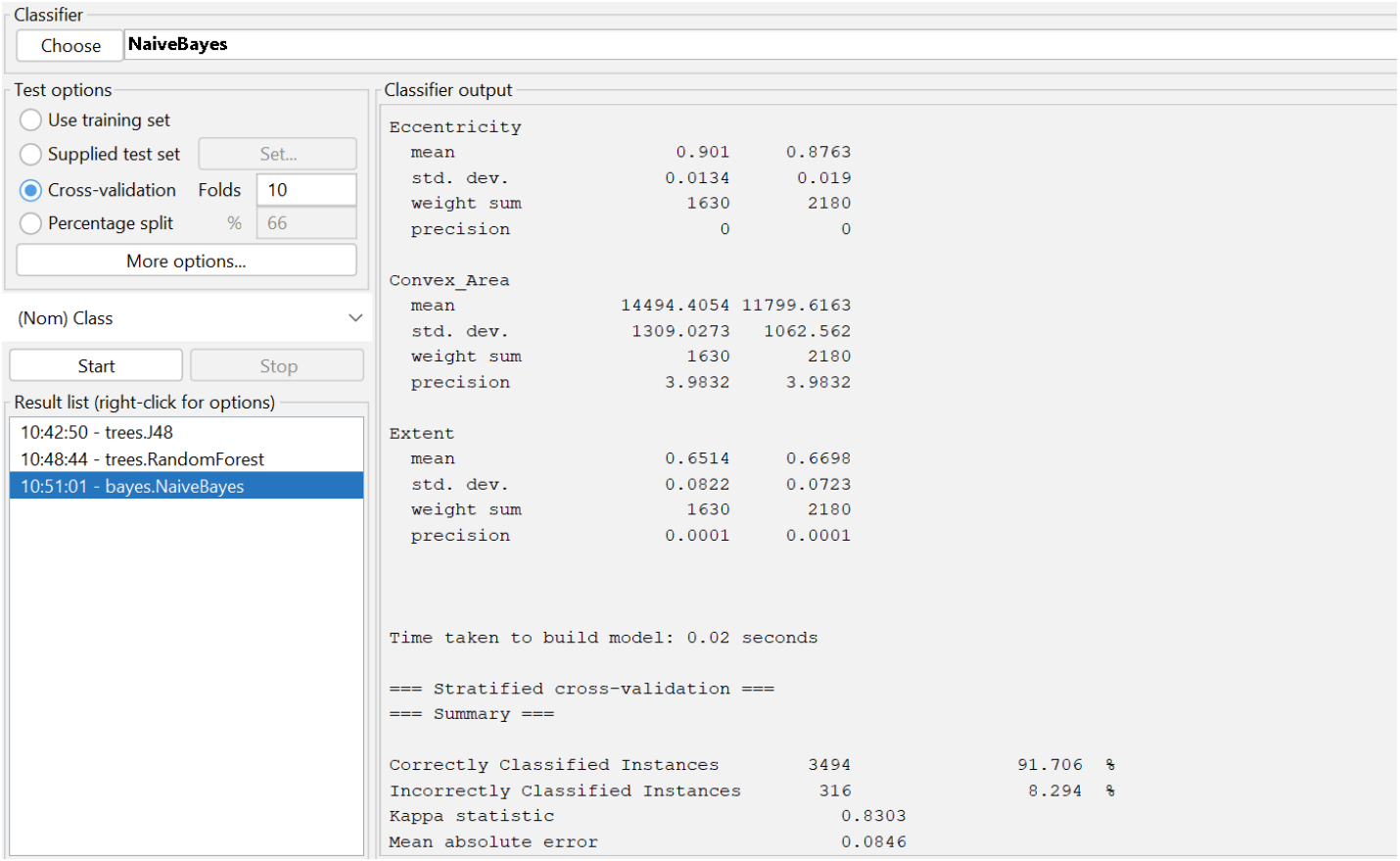


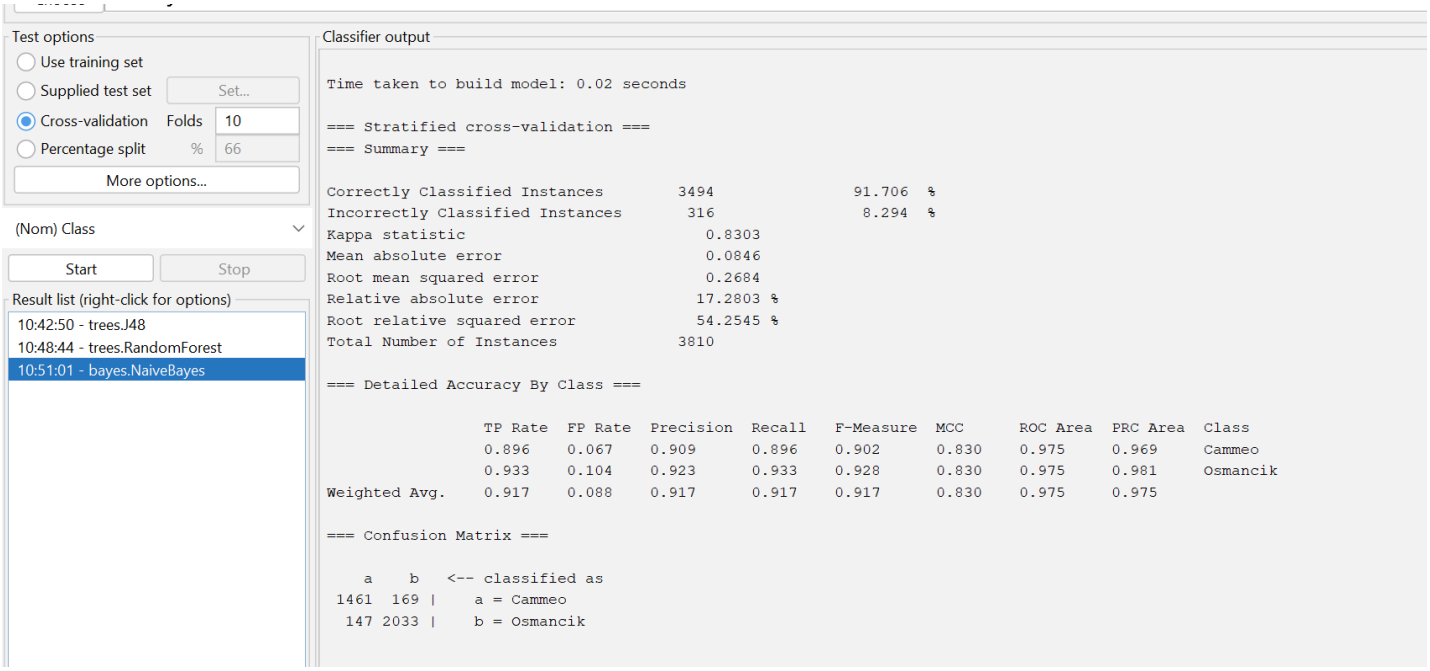

Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
10:42:50 - trees.J48
10:48:44 - trees.RandomForest
Classifier output
Time taken to build model: 0.79 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 3522 92.4409 %
Incorrectly Classified Instances 288 7.5591 %
Kappa statistic 0.8454
Mean absolute error 0.1099
Root mean squared error 0.2435
Relative absolute error 22.4572 %
Root relative squared error 49.2227 %
Total Number of Instances 3810
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.906 0.062 0.916 0.906 0.911 0.845 0.973 0.963 Cammeo
0.938 0.094 0.930 0.938 0.934 0.845 0.973 0.977 Osmancik
Weighted Avg. 0.924 0.080 0.924 0.924 0.924 0.845 0.973 0.971
=== Confusion Matrix ===
a b <-- classified as
1477 153 | a = Cammeo
135 2045 | b = Osmancik

Naive Bayes

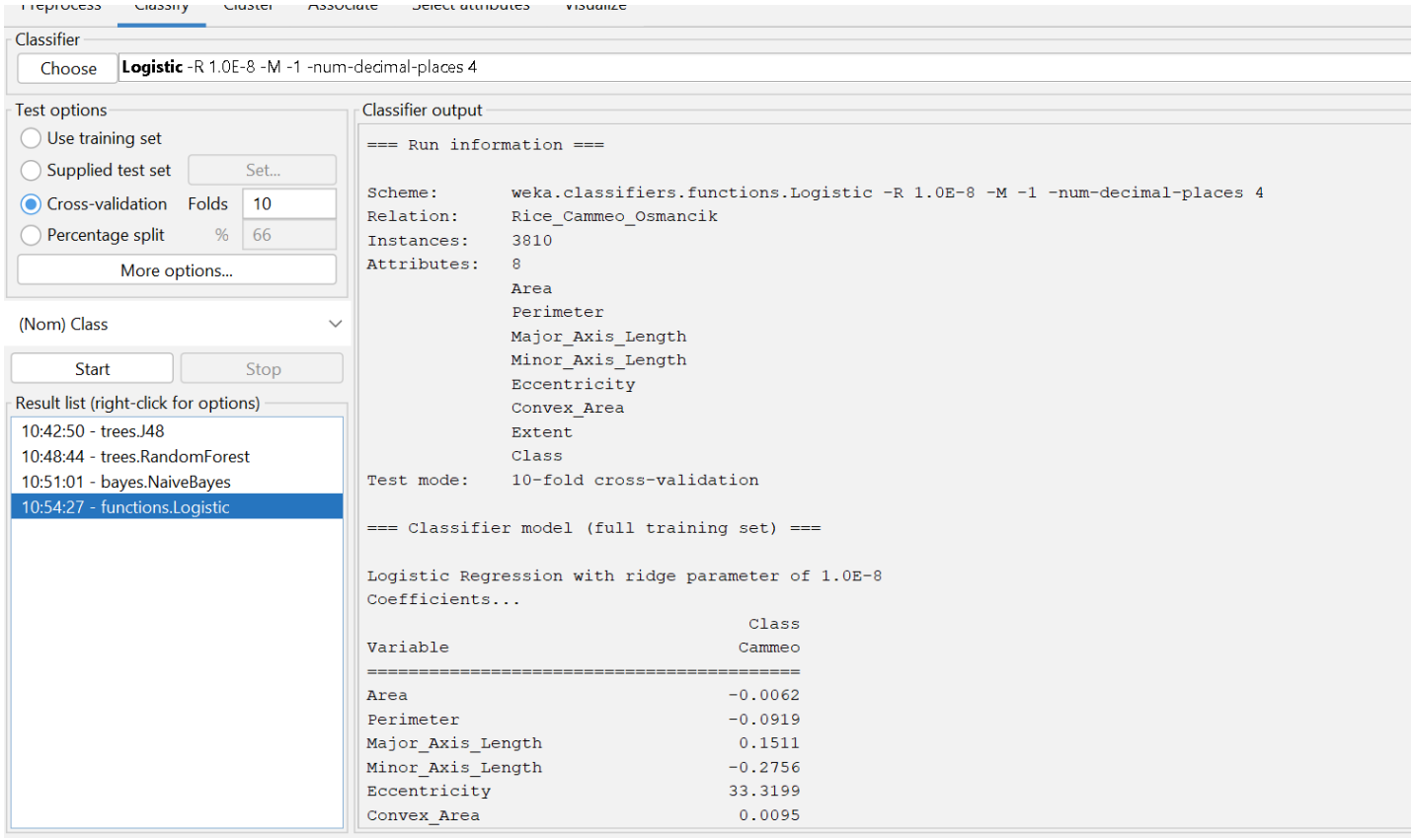

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
10:42:50 - trees.J48
10:48:44 - trees.RandomForest
10:51:01 - bayes.NaiveBayes
Classifier output
=== Run information ===
Scheme: weka.classifiers.bayes.NaiveBayes
Relation: Rice_Cammeo_Osmancik
Instances: 3810
Attributes: 8
Area
Perimeter
Major_Axis_Length
Minor_Axis_Length
Eccentricity
Convex_Area
Extent
Class
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
Naive Bayes Classifier
Class
Attribute Cammeo Osmancik
(0.43) (0.57)

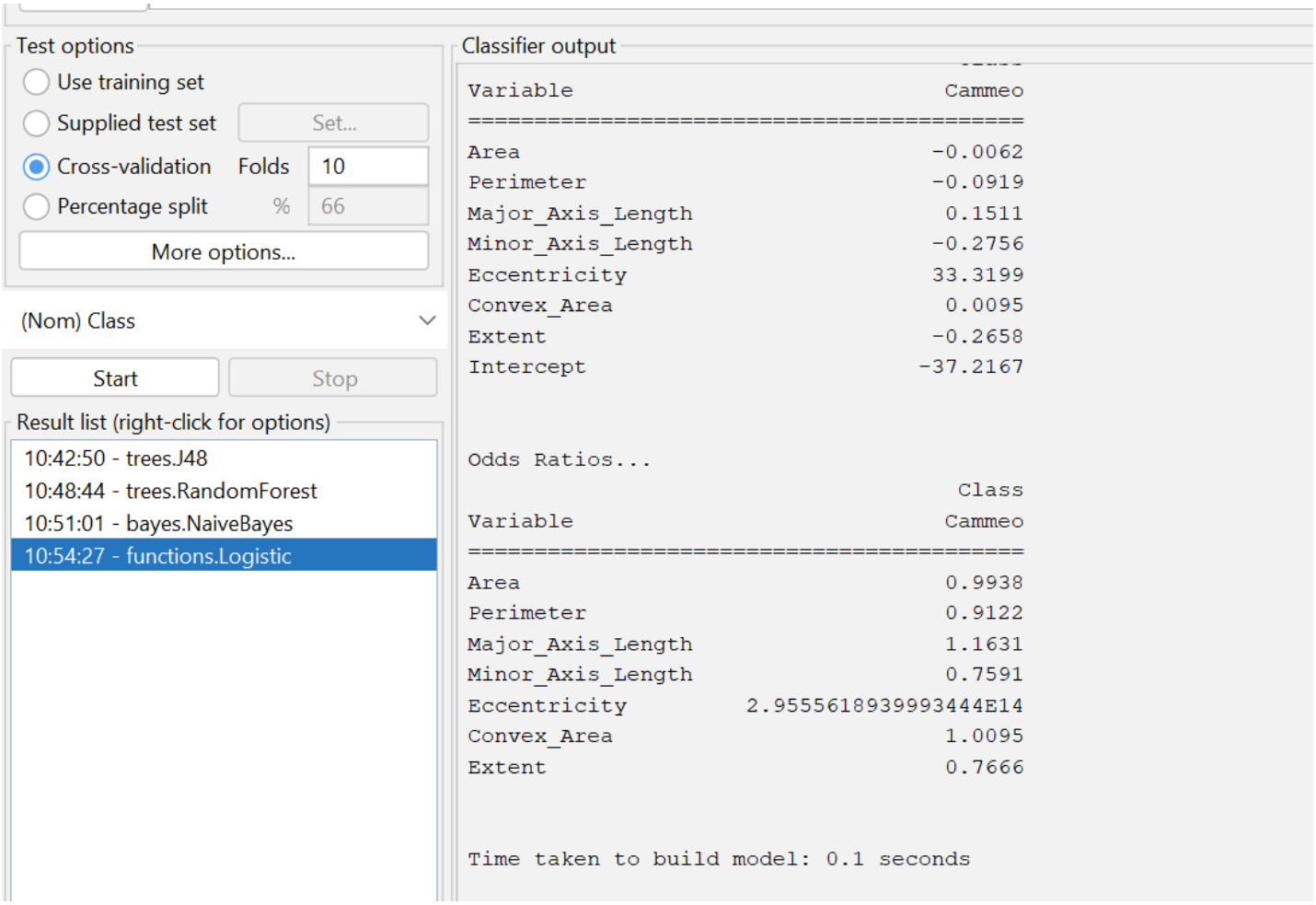

Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
10:42:50 - trees.J48
10:48:44 - trees.RandomForest
10:51:01 - bayes.NaiveBayes
Classifier output
Attribute Cammeo Osmancik
(0.43) (0.57)
Area
mean 14162.9242 11549.8248
std. dev. 1286.3799 1041.7108
weight sum 1630 2180
precision 4.0191 4.0191
Perimeter
mean 487.4392 429.4153
std. dev. 22.1746 20.1494
weight sum 1630 2180
precision 0.0507 0.0507
Major_Axis_Length
mean 205.4791 176.2877
std. dev. 10.3308 9.3602
weight sum 1630 2180
precision 0.0246 0.0246
Minor_Axis_Length
mean 88.7675 84.479
std. dev. 5.3485 5.3014
weight sum 1630 2180
precision 0.0126 0.0126

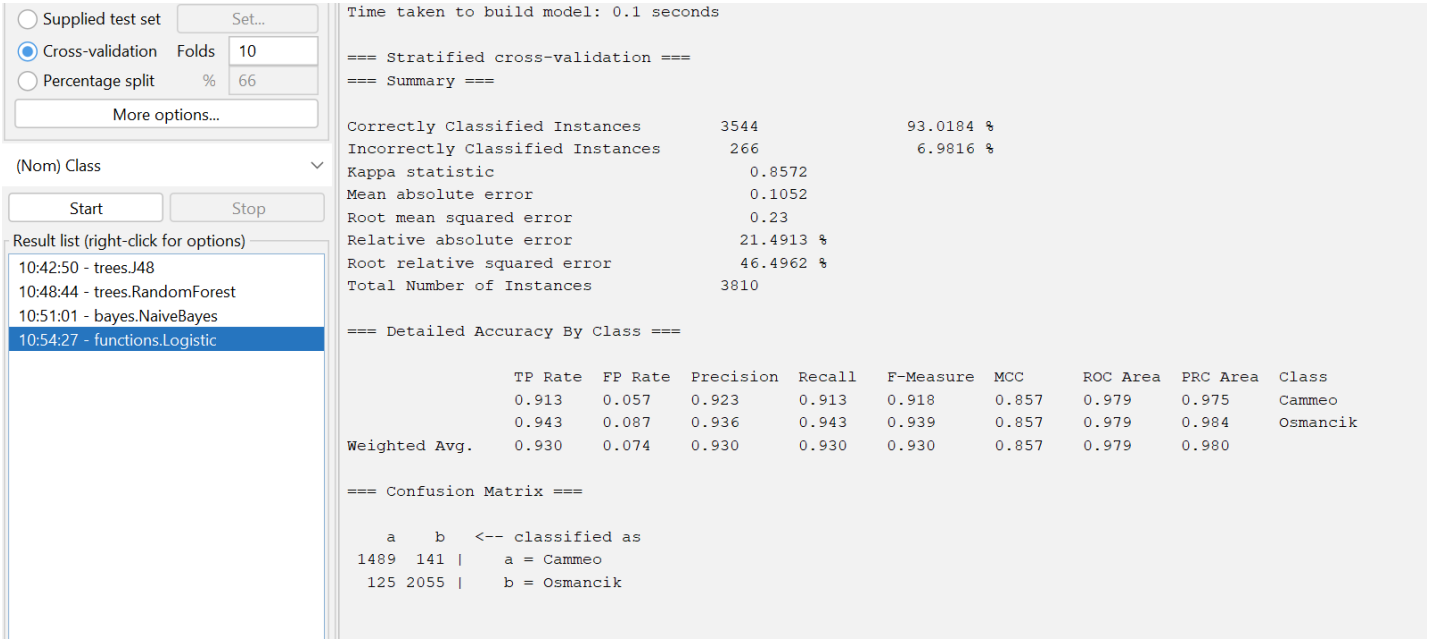

Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
10:42:50 - trees.J48
10:48:44 - trees.RandomForest
10:51:01 - bayes.NaiveBayes
Classifier output
Eccentricity
mean 0.901 0.8763
std. dev. 0.0134 0.019
weight sum 1630 2180
precision 0 0
Convex_Area
mean 14494.4054 11799.6163
std. dev. 1309.0273 1062.562
weight sum 1630 2180
precision 3.9832 3.9832
Extent
mean 0.6514 0.6698
std. dev. 0.0822 0.0723
weight sum 1630 2180
precision 0.0001 0.0001
Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 3494 91.706 %
Incorrectly Classified Instances 316 8.294 %
Kappa statistic 0.8303
Mean absolute error 0.0846


Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
10:42:50 - trees.J48
10:48:44 - trees.RandomForest
10:51:01 - bayes.NaiveBayes
Classifier output
Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 3494 91.706 %
Incorrectly Classified Instances 316 8.294 %
Kappa statistic 0.8303
Mean absolute error 0.0846
Root mean squared error 0.2684
Relative absolute error 17.2803 %
Root relative squared error 54.2545 %
Total Number of Instances 3810
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.896 0.067 0.909 0.896 0.902 0.830 0.975 0.969 Cammeo
0.933 0.104 0.923 0.933 0.928 0.830 0.975 0.981 Osmancik
Weighted Avg. 0.917 0.088 0.917 0.917 0.917 0.830 0.975 0.975
=== Confusion Matrix ===
a b <-- classified as
1461 169 | a = Cammeo
147 2033 | b = Osmancik

Logistic Regression




Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
10:42:50 - trees.J48
10:48:44 - trees.RandomForest
10:51:01 - bayes.NaiveBayes
10:54:27 - functions.Logistic
Classifier output
Variable Cammeo
Area -0.0062
Perimeter -0.0919
Major_Axis_Length 0.1511
Minor_Axis_Length -0.2756
Eccentricity 33.3199
Convex_Area 0.0095
Extent -0.2658
Intercept -37.2167
Odds Ratios...
Class
Variable Cammeo
Area 0.9938
Perimeter 0.9122
Major_Axis_Length 1.1631
Minor_Axis_Length 0.7591
Eccentricity 2.9555618939993444E14
Convex_Area 1.0095
Extent 0.7666
Time taken to build model: 0.1 seconds


Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
10:42:50 - trees.J48
10:48:44 - trees.RandomForest
10:51:01 - bayes.NaiveBayes
10:54:27 - functions.Logistic
Time taken to build model: 0.1 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 3544 93.0184 %
Incorrectly Classified Instances 266 6.9816 %
Kappa statistic 0.8572
Mean absolute error 0.1052
Root mean squared error 0.23
Relative absolute error 21.4913 %
Root relative squared error 46.4962 %
Total Number of Instances 3810
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.913 0.057 0.923 0.913 0.918 0.857 0.979 0.975 Cammeo
0.943 0.087 0.936 0.943 0.939 0.857 0.979 0.984 Osmancik
Weighted Avg. 0.930 0.074 0.930 0.930 0.930 0.857 0.979 0.980
=== Confusion Matrix ===
a b <-- classified as
1489 141 | a = Cammeo
125 2055 | b = Osmancik

2.1.4 results

Table 1: Overall Performance

Model	Accuracy (%)	Kappa
J48	93.02	0.8572
Random Forest	92.49	0.8467
Logistic Regression	92.44	0.8454
Naive Bayes	91.71	0.8303

Table 2: Best Model (J48)

Class	Precision	Recall	F1
Cammeo	0.923	0.913	0.918
Osmancik	0.936	0.943	0.939

Table 3: Error Comparison

Model	MAE	RMSE
J48	0.1052	0.2300
Random Forest	0.1320	0.2597
Logistic Regression	0.1099	0.2435
Naive Bayes	0.0846	0.2684

2.2 Dataset 2: Multi-class Classification – Car Evaluation

2.2.1 Description

The Car Evaluation dataset presents a **multi-class classification** problem to predict a car's acceptability based on six categorical attributes related to cost and technical characteristics. Derived from a hierarchical decision model, the goal is to classify a car into one of four acceptability levels: **unacceptable, acceptable, good, or very good**. This dataset is often used to test rule-learning and decision tree algorithms due to its clear, structured relationship between simple features and the target class.

2.2.2 Features

All six predictive features are **categorical**. The target variable has four ordered classes.

Feature Name	Data Type	Description	Possible Values
buying	Categorical	Initial purchase price.	vhigh, high, med, low
maint	Categorical	Price of maintenance.	vhigh, high, med, low
doors	Categorical	Number of doors.	2, 3, 4, 5more
persons	Categorical	Passenger capacity.	2, 4, more
lug_boot	Categorical	Size of the luggage boot (trunk).	small, med, big

safety	Categorical	Estimated safety rating of the car.	low, med, high
class (Target)	Categorical	Final evaluation of car acceptability.	unacc, acc, good, vgood

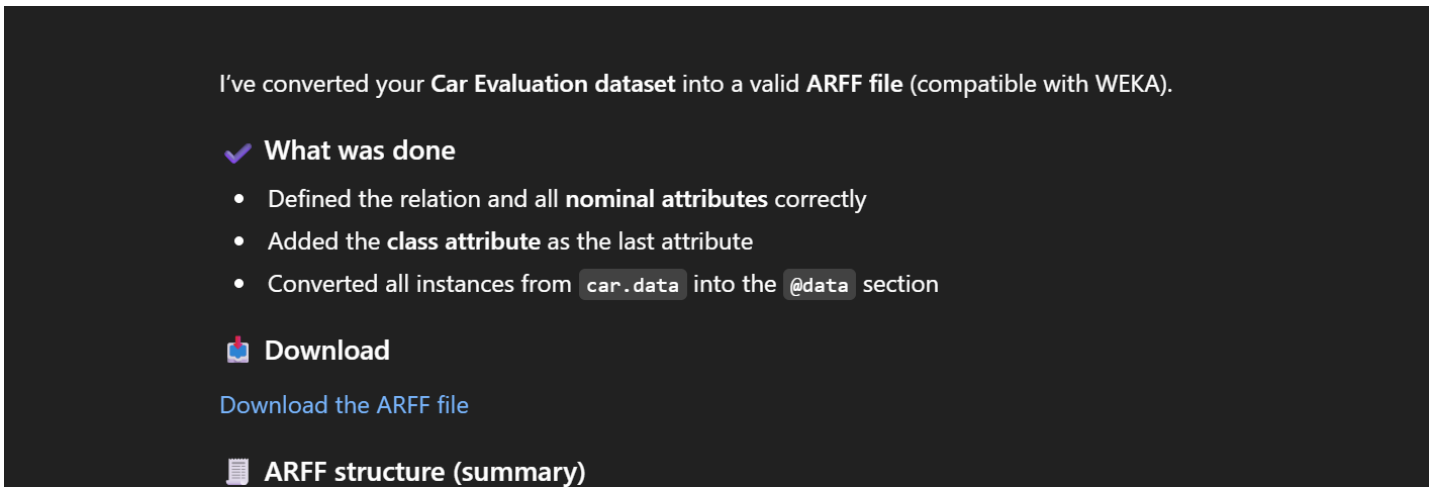
Key Dataset Facts:

- **Number of Instances:** 1,728
- **Number of Features:** 6 (predictive) + 1 (target)
- **Missing Values:** None
- **Class Distribution:** Heavily skewed towards the unacceptable class.

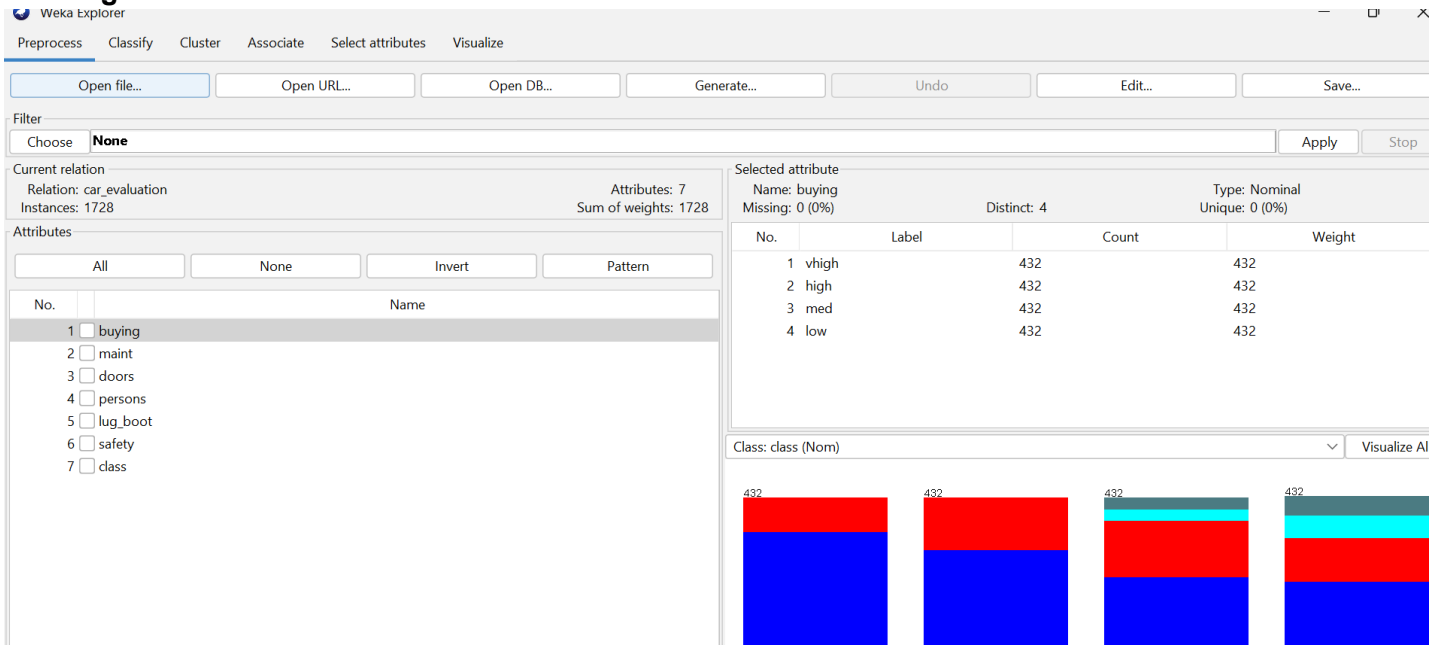
2.2.3 Data Preparation for WEKA

The dataset required conversion from its native data file to the ARFF format. This was done using WEKA's built-in CSV loader or by manually creating an ARFF header that defines the attributes and their nominal values as listed above. All features are symbolic, so no scaling was necessary. The class imbalance is noted for consideration during model evaluation.

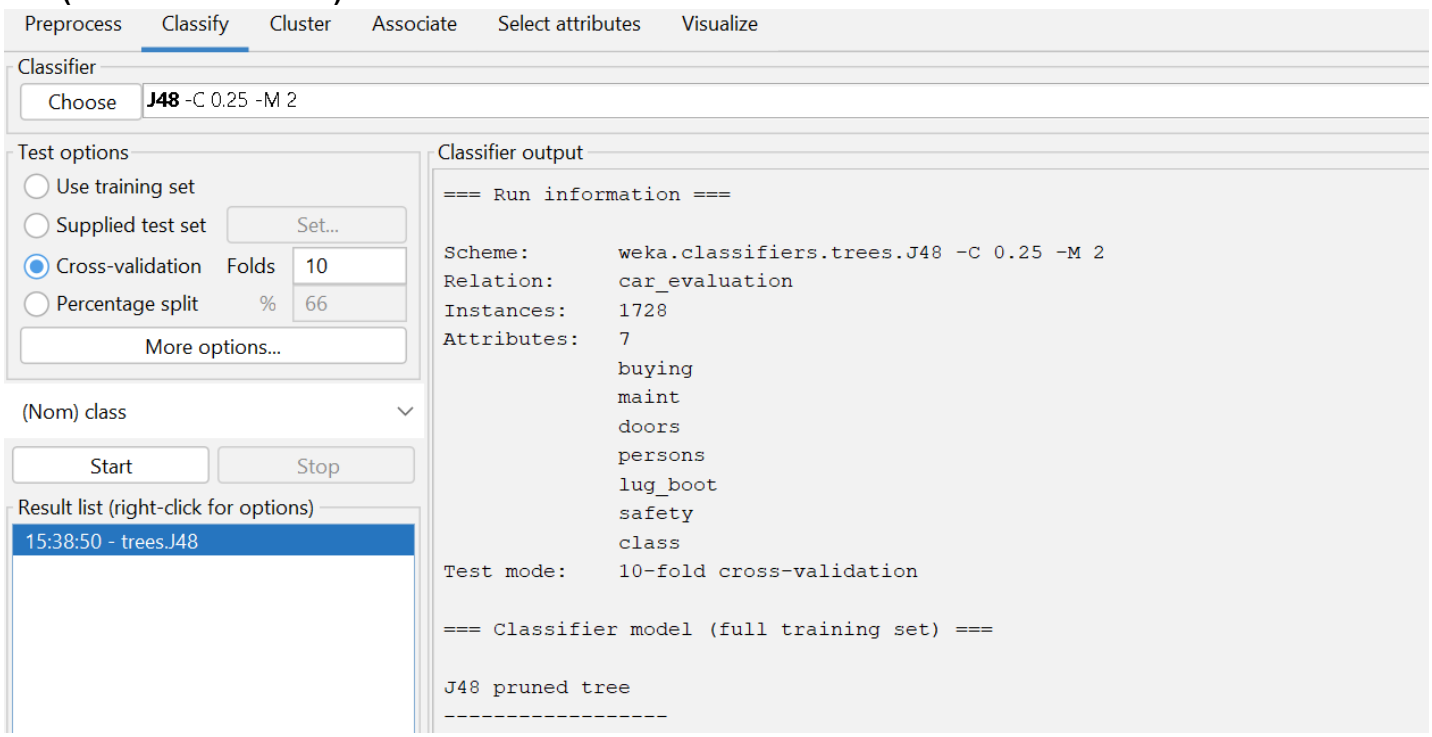
Converting to arff using chatgpt

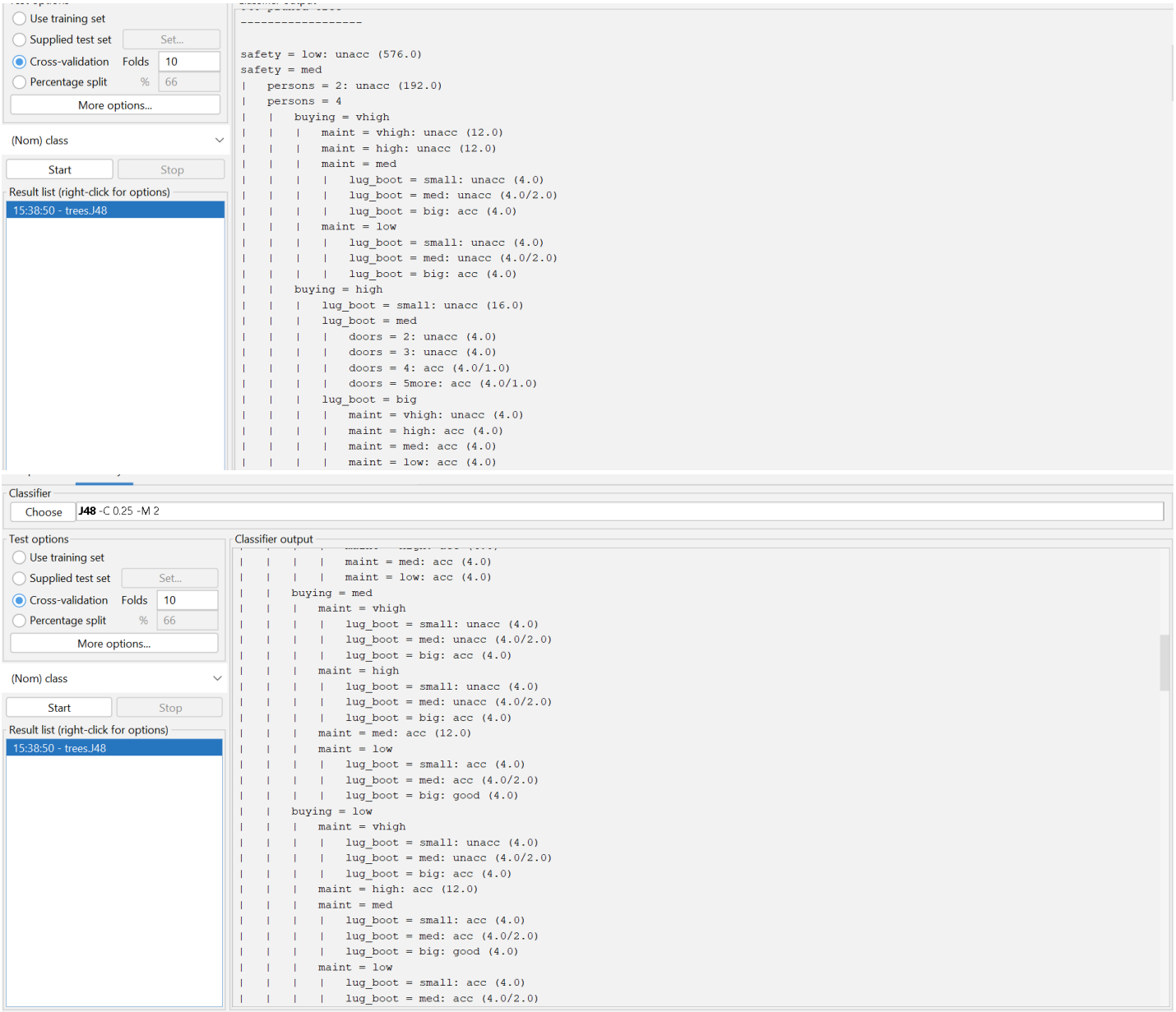


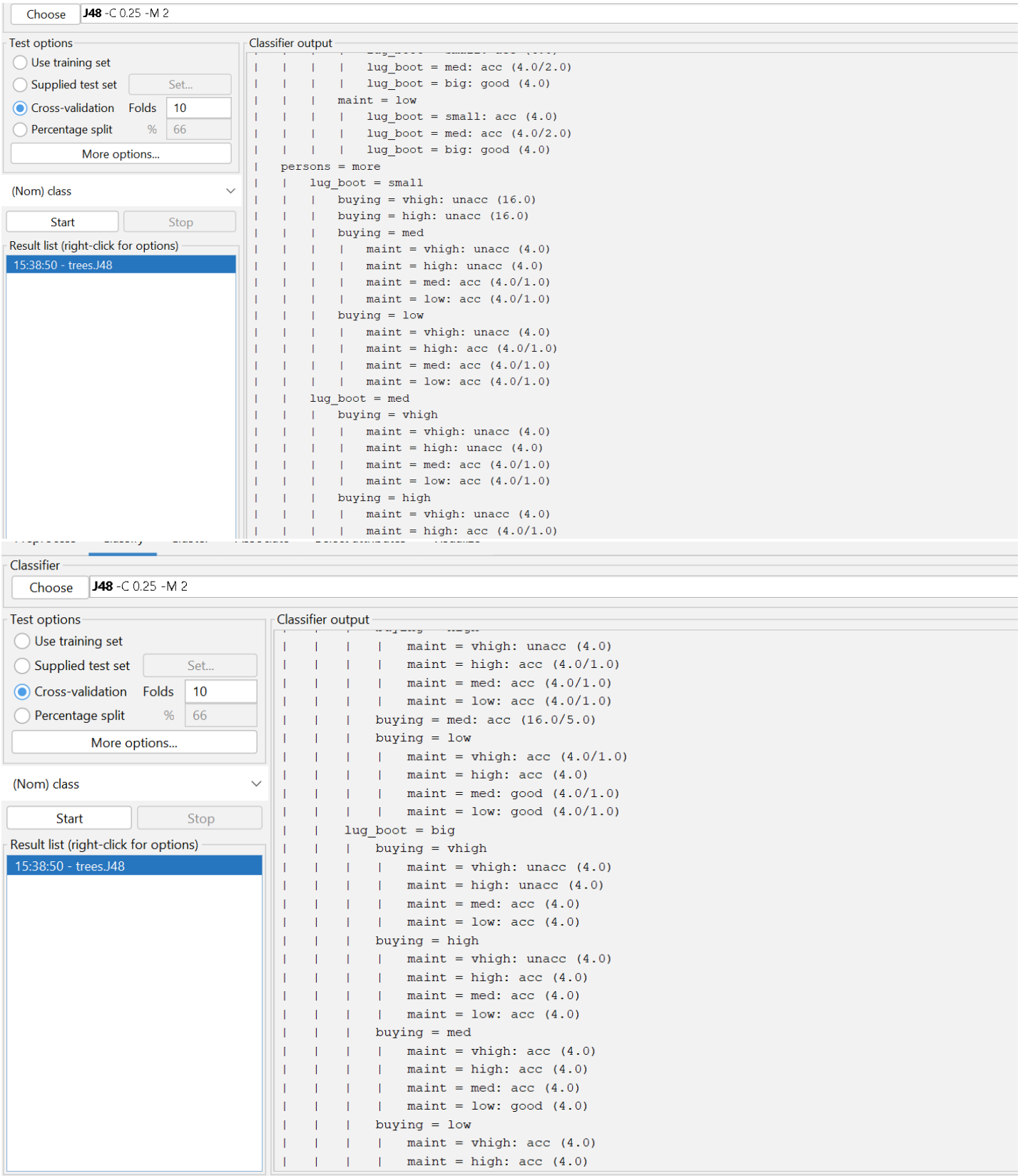
Loading data

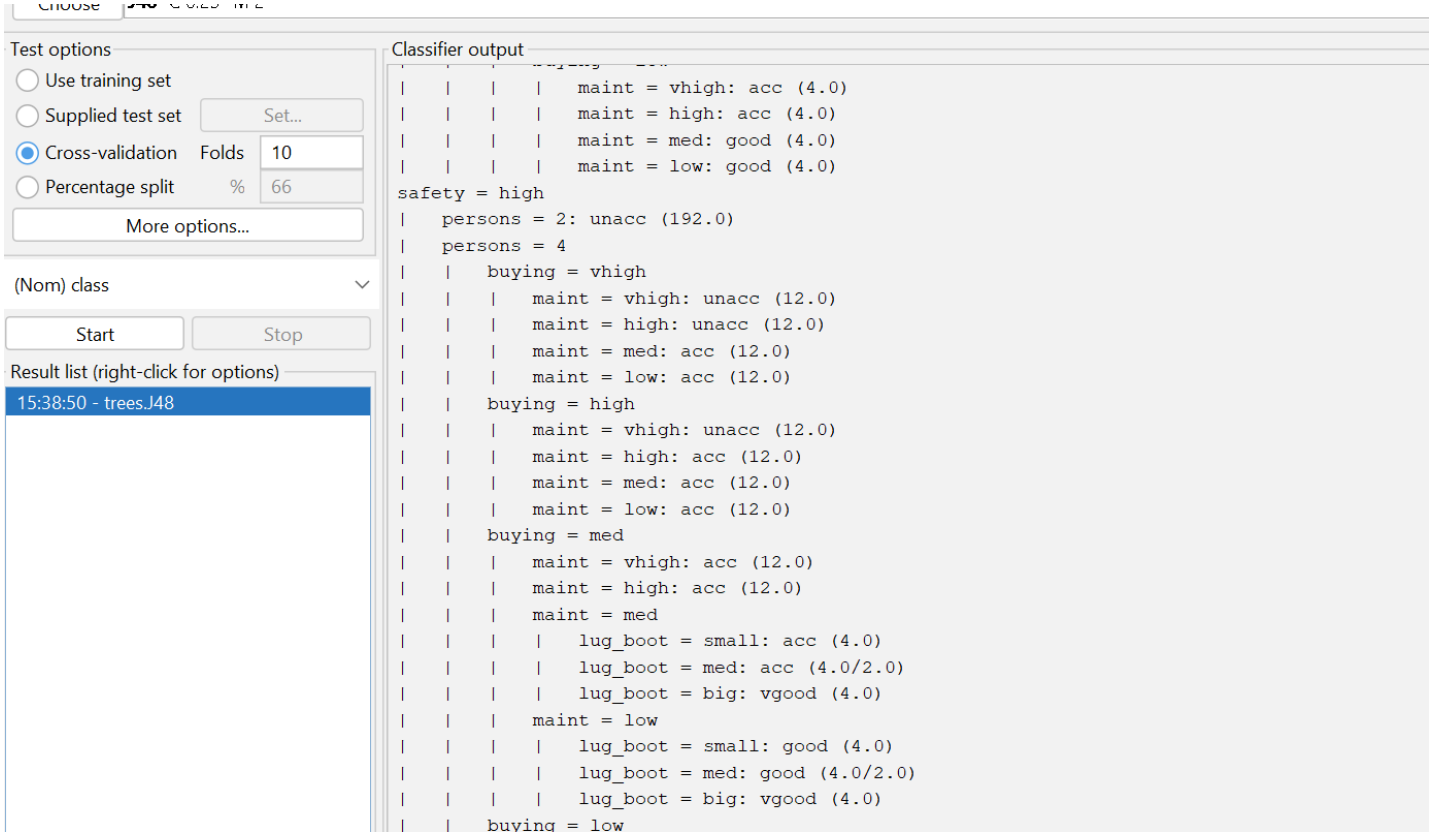


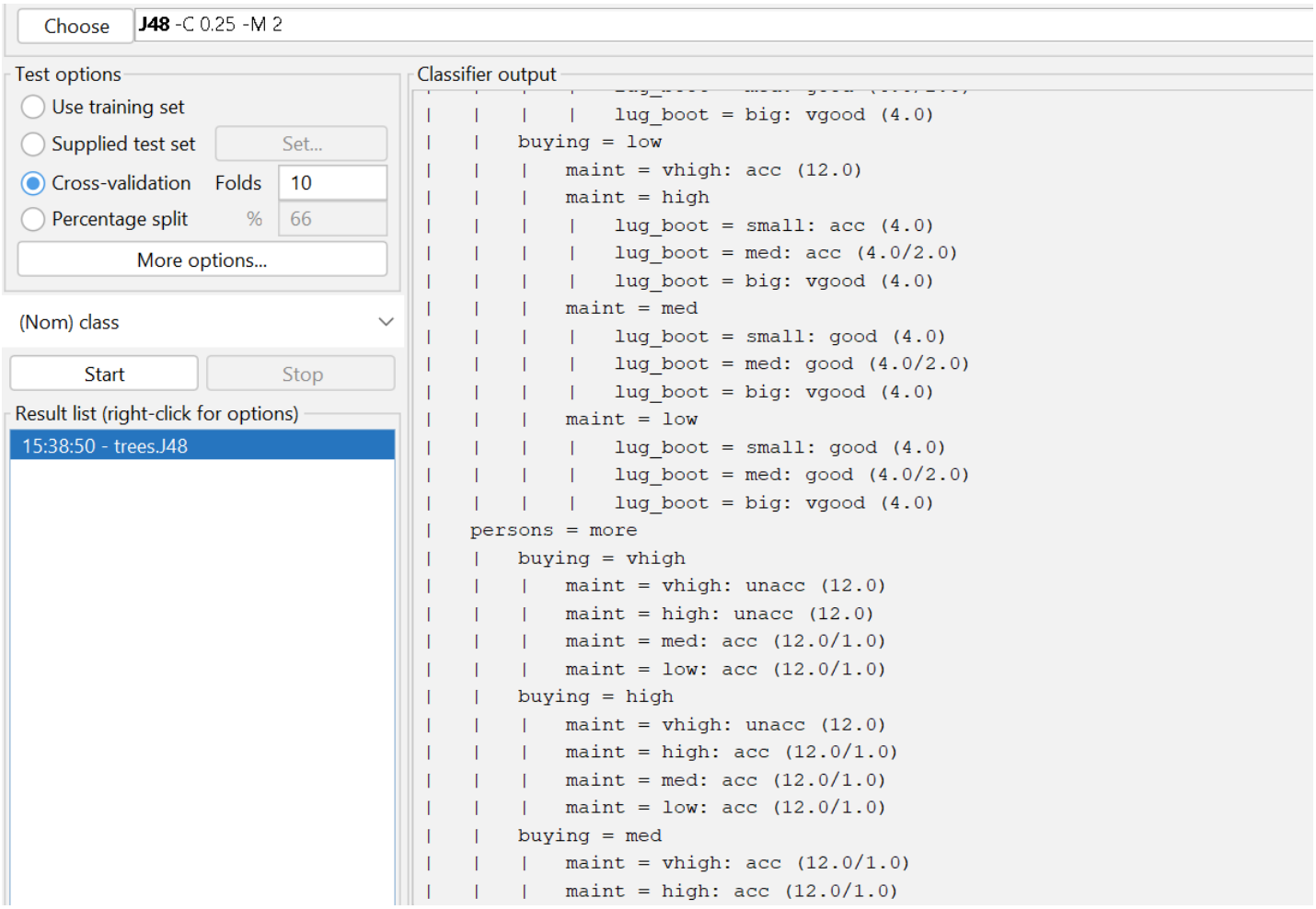
J48 (C4.5 Decision Tree)

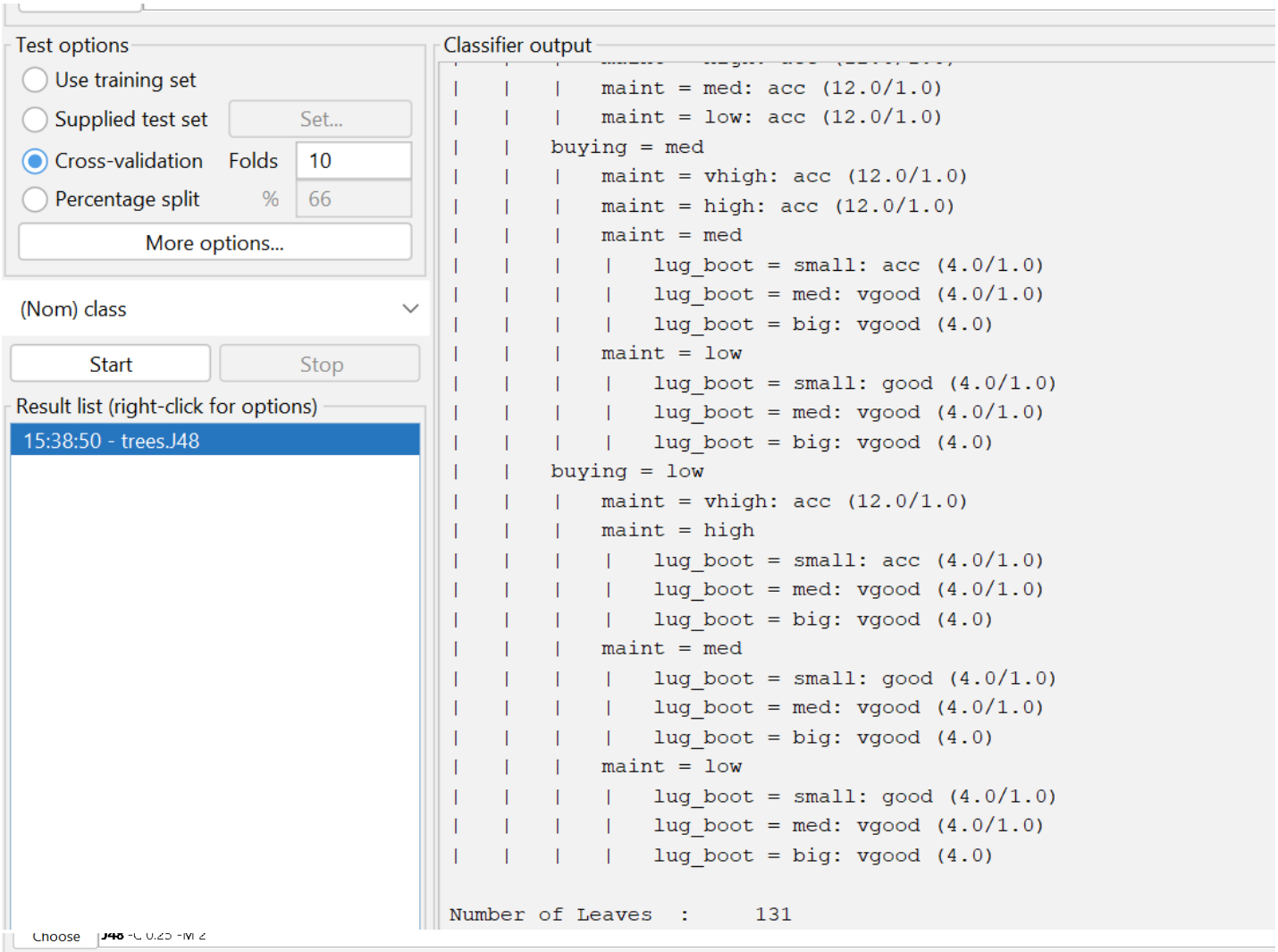


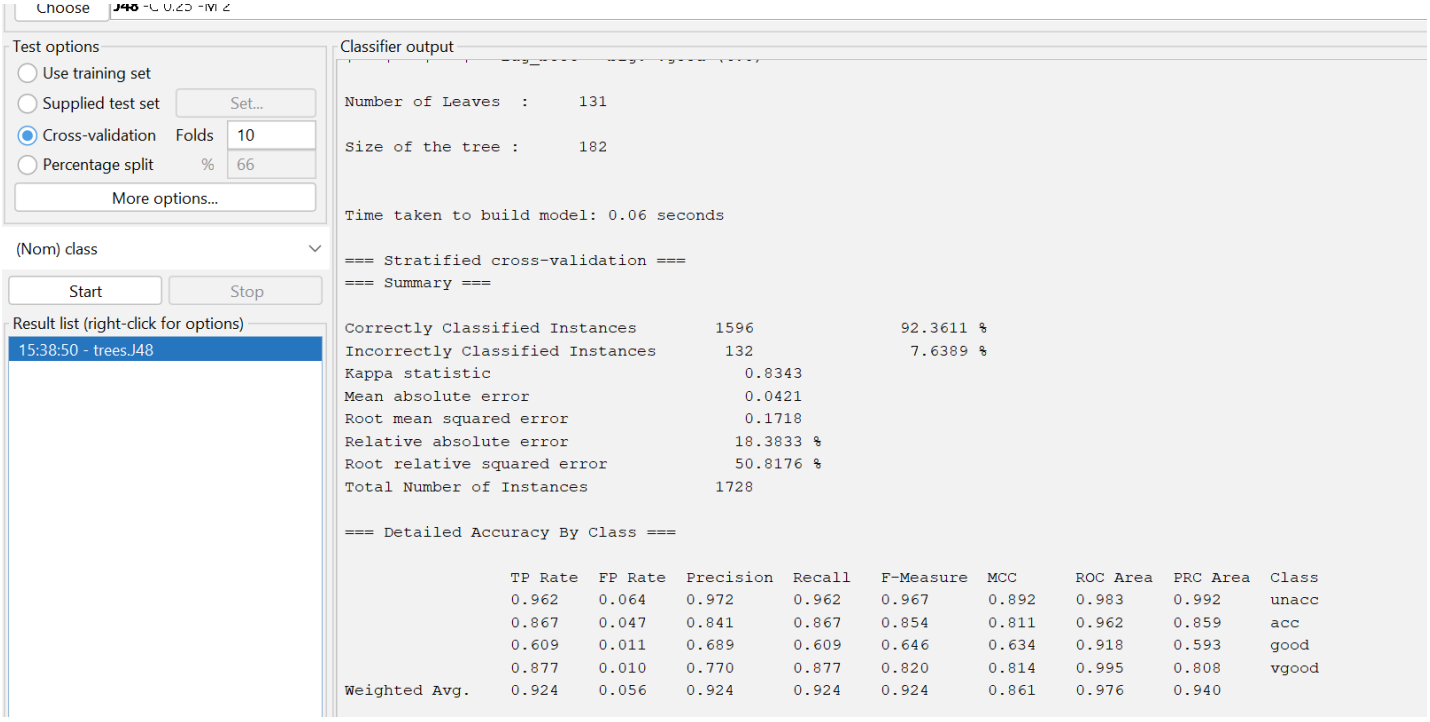

Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
Classifier output
safety = low: unacc (576.0)
safety = med
| persons = 2: unacc (192.0)
| persons = 4
| | buying = vhigh
| | | maint = vhigh: unacc (12.0)
| | | maint = high: unacc (12.0)
| | | maint = med
| | | | lug_boot = small: unacc (4.0)
| | | | lug_boot = med: unacc (4.0/2.0)
| | | | lug_boot = big: acc (4.0)
| | | maint = low
| | | | lug_boot = small: unacc (4.0)
| | | | lug_boot = med: unacc (4.0/2.0)
| | | | lug_boot = big: acc (4.0)
| | buying = high
| | | lug_boot = small: unacc (16.0)
| | | lug_boot = med
| | | | doors = 2: unacc (4.0)
| | | | doors = 3: unacc (4.0)
| | | | doors = 4: acc (4.0/1.0)
| | | | doors = 5more: acc (4.0/1.0)
| | | lug_boot = big
| | | | maint = vhigh: unacc (4.0)
| | | | maint = high: acc (4.0)
| | | | maint = med: acc (4.0)
| | | | maint = low: acc (4.0)
| | | | maint = med: acc (4.0)
| | | | maint = low: acc (4.0)
| | buying = med
| | | maint = vhigh
| | | | lug_boot = small: unacc (4.0)
| | | | lug_boot = med: unacc (4.0/2.0)
| | | | lug_boot = big: acc (4.0)
| | | maint = high
| | | | lug_boot = small: unacc (4.0)
| | | | lug_boot = med: unacc (4.0/2.0)
| | | | lug_boot = big: acc (4.0)
| | | maint = med: acc (12.0)
| | | maint = low
| | | | lug_boot = small: acc (4.0)
| | | | lug_boot = med: acc (4.0/2.0)
| | | | lug_boot = big: good (4.0)
| | buying = low
| | | maint = vhigh
| | | | lug_boot = small: unacc (4.0)
| | | | lug_boot = med: unacc (4.0/2.0)
| | | | lug_boot = big: acc (4.0)
| | | maint = high: acc (12.0)
| | | maint = med
| | | | lug_boot = small: acc (4.0)
| | | | lug_boot = med: acc (4.0/2.0)
| | | | lug_boot = big: good (4.0)
| | | maint = low
| | | | lug_boot = small: acc (4.0)
| | | | lug_boot = med: acc (4.0/2.0)

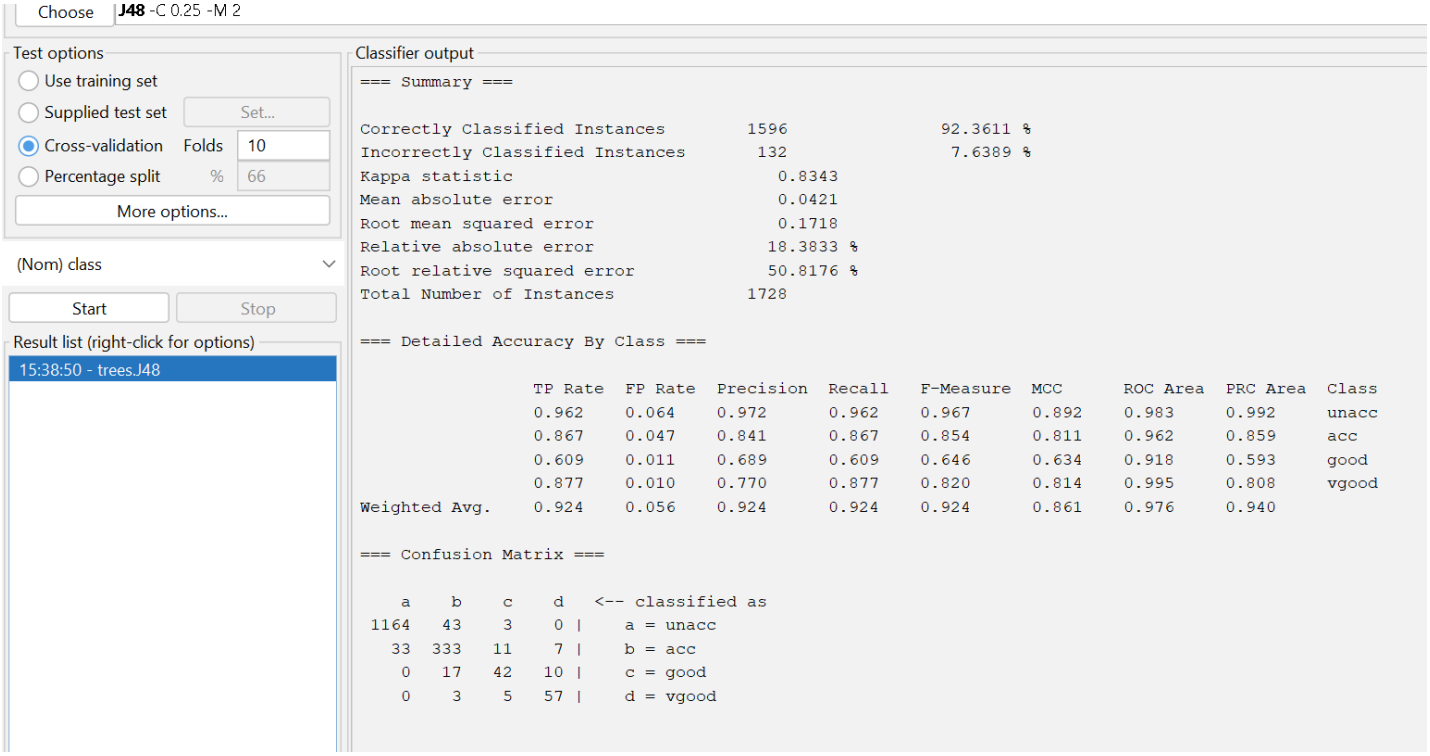

Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
Classifier output
| | | | lug_boot = med: acc (4.0/2.0)
| | | | lug_boot = big: good (4.0)
| | | maint = low
| | | | lug_boot = small: acc (4.0)
| | | | lug_boot = med: acc (4.0/2.0)
| | | | lug_boot = big: good (4.0)
| persons = more
| | lug_boot = small
| | | buying = vhigh: unacc (16.0)
| | | buying = high: unacc (16.0)
| | | buying = med
| | | | maint = vhigh: unacc (4.0)
| | | | maint = high: unacc (4.0)
| | | | maint = med: acc (4.0/1.0)
| | | | maint = low: acc (4.0/1.0)
| | | buying = low
| | | | maint = vhigh: unacc (4.0)
| | | | maint = high: acc (4.0/1.0)
| | | | maint = med: acc (4.0/1.0)
| | | | maint = low: acc (4.0/1.0)
| | lug_boot = med
| | | buying = vhigh
| | | | maint = vhigh: unacc (4.0)
| | | | maint = high: unacc (4.0)
| | | | maint = med: acc (4.0/1.0)
| | | | maint = low: acc (4.0/1.0)
| | | buying = high
| | | | maint = vhigh: unacc (4.0)
| | | | maint = high: acc (4.0/1.0)
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
Classifier output
| | | | maint = vhigh: unacc (4.0)
| | | | maint = high: acc (4.0/1.0)
| | | | maint = med: acc (4.0/1.0)
| | | | maint = low: acc (4.0/1.0)
| | | buying = med: acc (16.0/5.0)
| | | buying = low
| | | | maint = vhigh: acc (4.0/1.0)
| | | | maint = high: acc (4.0)
| | | | maint = med: good (4.0/1.0)
| | | | maint = low: good (4.0/1.0)
| | lug_boot = big
| | | buying = vhigh
| | | | maint = vhigh: unacc (4.0)
| | | | maint = high: unacc (4.0)
| | | | maint = med: acc (4.0)
| | | | maint = low: acc (4.0)
| | | buying = high
| | | | maint = vhigh: unacc (4.0)
| | | | maint = high: acc (4.0)
| | | | maint = med: acc (4.0)
| | | | maint = low: acc (4.0)
| | | buying = med
| | | | maint = vhigh: acc (4.0)
| | | | maint = high: acc (4.0)
| | | | maint = med: acc (4.0)
| | | | maint = low: good (4.0)
| | | buying = low
| | | | maint = vhigh: acc (4.0)
| | | | maint = high: acc (4.0)


Choose
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
Classifier output
| | | | maint = vhigh: acc (4.0)
| | | | maint = high: acc (4.0)
| | | | maint = med: good (4.0)
| | | | maint = low: good (4.0)
safety = high
| persons = 2: unacc (192.0)
| persons = 4
| | buying = vhigh
| | | maint = vhigh: unacc (12.0)
| | | maint = high: unacc (12.0)
| | | maint = med: acc (12.0)
| | | maint = low: acc (12.0)
| | buying = high
| | | maint = vhigh: unacc (12.0)
| | | maint = high: acc (12.0)
| | | maint = med: acc (12.0)
| | | maint = low: acc (12.0)
| | buying = med
| | | maint = vhigh: acc (12.0)
| | | maint = high: acc (12.0)
| | | maint = med
| | | | lug_boot = small: acc (4.0)
| | | | lug_boot = med: acc (4.0/2.0)
| | | | lug_boot = big: vgood (4.0)
| | | maint = low
| | | | lug_boot = small: good (4.0)
| | | | lug_boot = med: good (4.0/2.0)
| | | | lug_boot = big: vgood (4.0)
| | buying = low

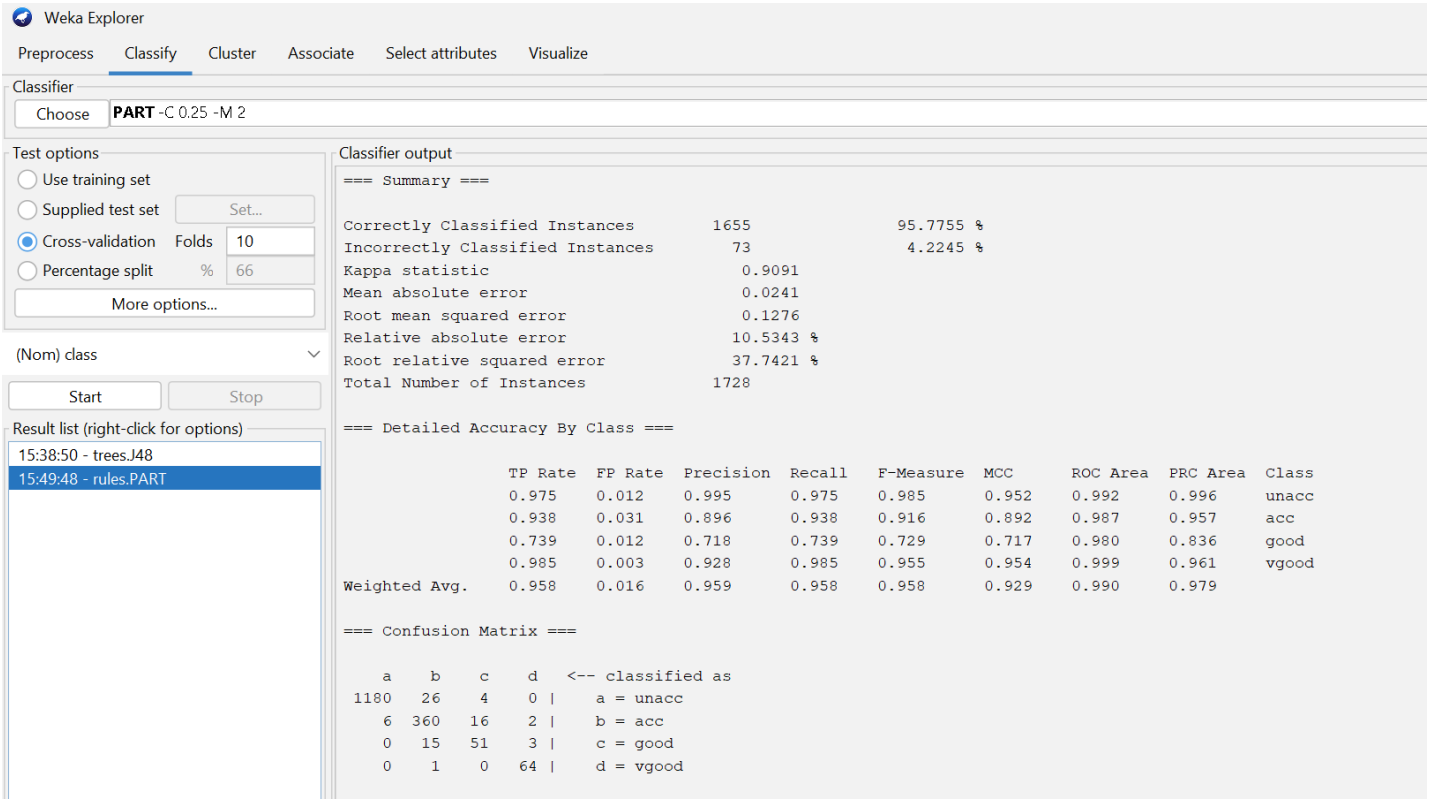

Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
Classifier output
| | | | lug_boot = big: vgood (4.0)
| | buying = low
| | | maint = vhigh: acc (12.0)
| | | maint = high
| | | | lug_boot = small: acc (4.0)
| | | | lug_boot = med: acc (4.0/2.0)
| | | | lug_boot = big: vgood (4.0)
| | | maint = med
| | | | lug_boot = small: good (4.0)
| | | | lug_boot = med: good (4.0/2.0)
| | | | lug_boot = big: vgood (4.0)
| | | maint = low
| | | | lug_boot = small: good (4.0)
| | | | lug_boot = med: good (4.0/2.0)
| | | | lug_boot = big: vgood (4.0)
| persons = more
| | buying = vhigh
| | | maint = vhigh: unacc (12.0)
| | | maint = high: unacc (12.0)
| | | maint = med: acc (12.0/1.0)
| | | maint = low: acc (12.0/1.0)
| | buying = high
| | | maint = vhigh: unacc (12.0)
| | | maint = high: acc (12.0/1.0)
| | | maint = med: acc (12.0/1.0)
| | | maint = low: acc (12.0/1.0)
| | buying = med
| | | maint = vhigh: acc (12.0/1.0)
| | | maint = high: acc (12.0/1.0)


Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
Classifier output
| | | maint = med: acc (12.0/1.0)
| | | maint = low: acc (12.0/1.0)
| | buying = med
| | | maint = vhigh: acc (12.0/1.0)
| | | maint = high: acc (12.0/1.0)
| | | maint = med
| | | | lug_boot = small: acc (4.0/1.0)
| | | | lug_boot = med: vgood (4.0/1.0)
| | | | lug_boot = big: vgood (4.0)
| | | maint = low
| | | | lug_boot = small: good (4.0/1.0)
| | | | lug_boot = med: vgood (4.0/1.0)
| | | | lug_boot = big: vgood (4.0)
| | buying = low
| | | maint = vhigh: acc (12.0/1.0)
| | | maint = high
| | | | lug_boot = small: acc (4.0/1.0)
| | | | lug_boot = med: vgood (4.0/1.0)
| | | | lug_boot = big: vgood (4.0)
| | | maint = med
| | | | lug_boot = small: good (4.0/1.0)
| | | | lug_boot = med: vgood (4.0/1.0)
| | | | lug_boot = big: vgood (4.0)
| | | maint = low
| | | | lug_boot = small: good (4.0/1.0)
| | | | lug_boot = med: vgood (4.0/1.0)
| | | | lug_boot = big: vgood (4.0)
Number of Leaves : 131
Choose


Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
Classifier output
Number of Leaves : 131
Size of the tree : 182
Time taken to build model: 0.06 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 1596 92.3611 %
Incorrectly Classified Instances 132 7.6389 %
Kappa statistic 0.8343
Mean absolute error 0.0421
Root mean squared error 0.1718
Relative absolute error 18.3833 %
Root relative squared error 50.8176 %
Total Number of Instances 1728
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.962 0.064 0.972 0.962 0.967 0.892 0.983 0.992 unacc
0.867 0.047 0.841 0.867 0.854 0.811 0.962 0.859 acc
0.609 0.011 0.689 0.609 0.646 0.634 0.918 0.593 good
0.877 0.010 0.770 0.877 0.820 0.814 0.995 0.808 vgood
Weighted Avg. 0.924 0.056 0.924 0.924 0.924 0.861 0.976 0.940


Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
Classifier output
=== Summary ===
Correctly Classified Instances 1596 92.3611 %
Incorrectly Classified Instances 132 7.6389 %
Kappa statistic 0.8343
Mean absolute error 0.0421
Root mean squared error 0.1718
Relative absolute error 18.3833 %
Root relative squared error 50.8176 %
Total Number of Instances 1728
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.962 0.064 0.972 0.962 0.967 0.892 0.983 0.992 unacc
0.867 0.047 0.841 0.867 0.854 0.811 0.962 0.859 acc
0.609 0.011 0.689 0.609 0.646 0.634 0.918 0.593 good
0.877 0.010 0.770 0.877 0.820 0.814 0.995 0.808 vgood
Weighted Avg. 0.924 0.056 0.924 0.924 0.924 0.861 0.976 0.940
=== Confusion Matrix ===
a b c d <-- classified as
1164 43 3 0 | a = unacc
33 333 11 7 | b = acc
0 17 42 10 | c = good
0 3 5 57 | d = vgood

PART (Rule-based classifier)

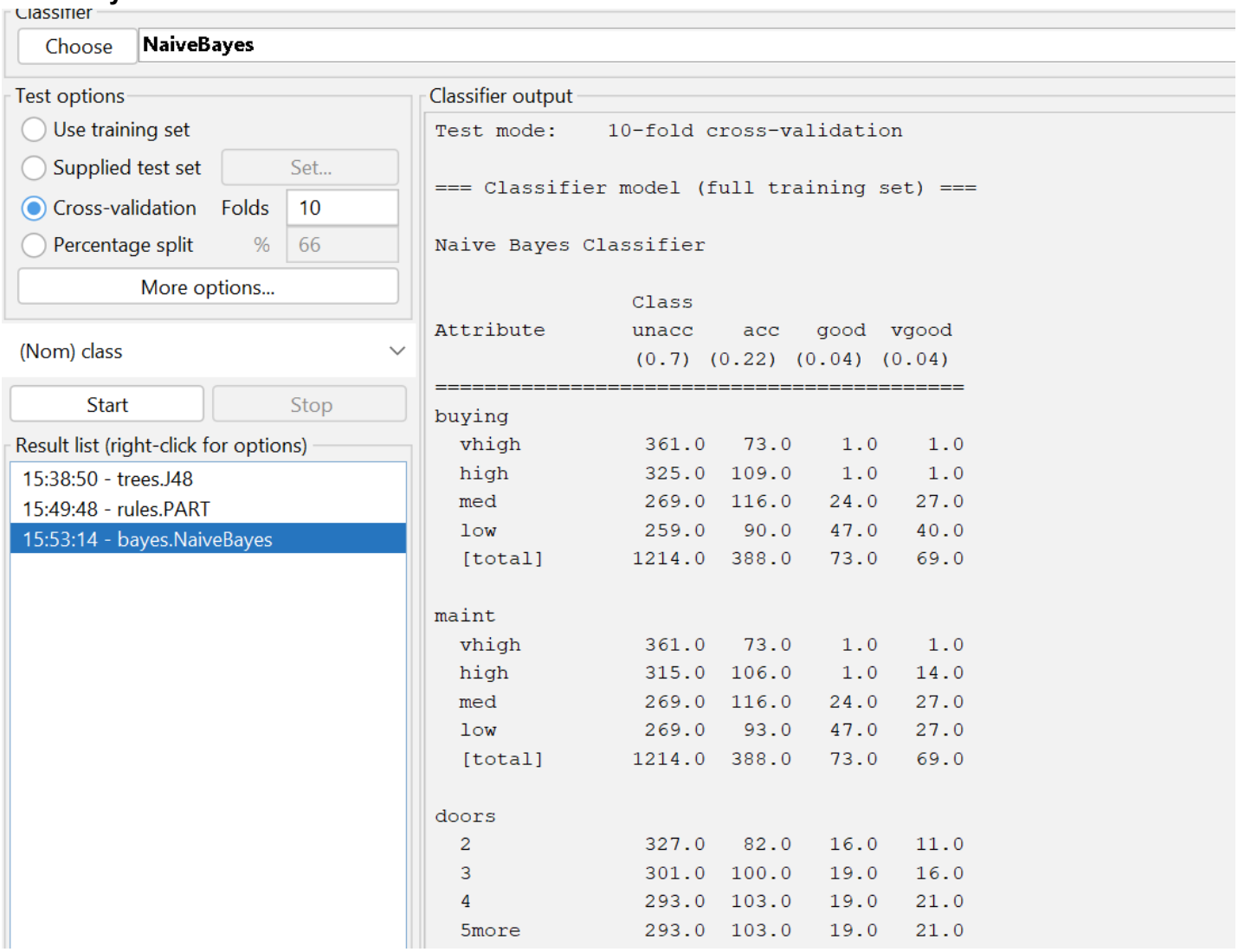


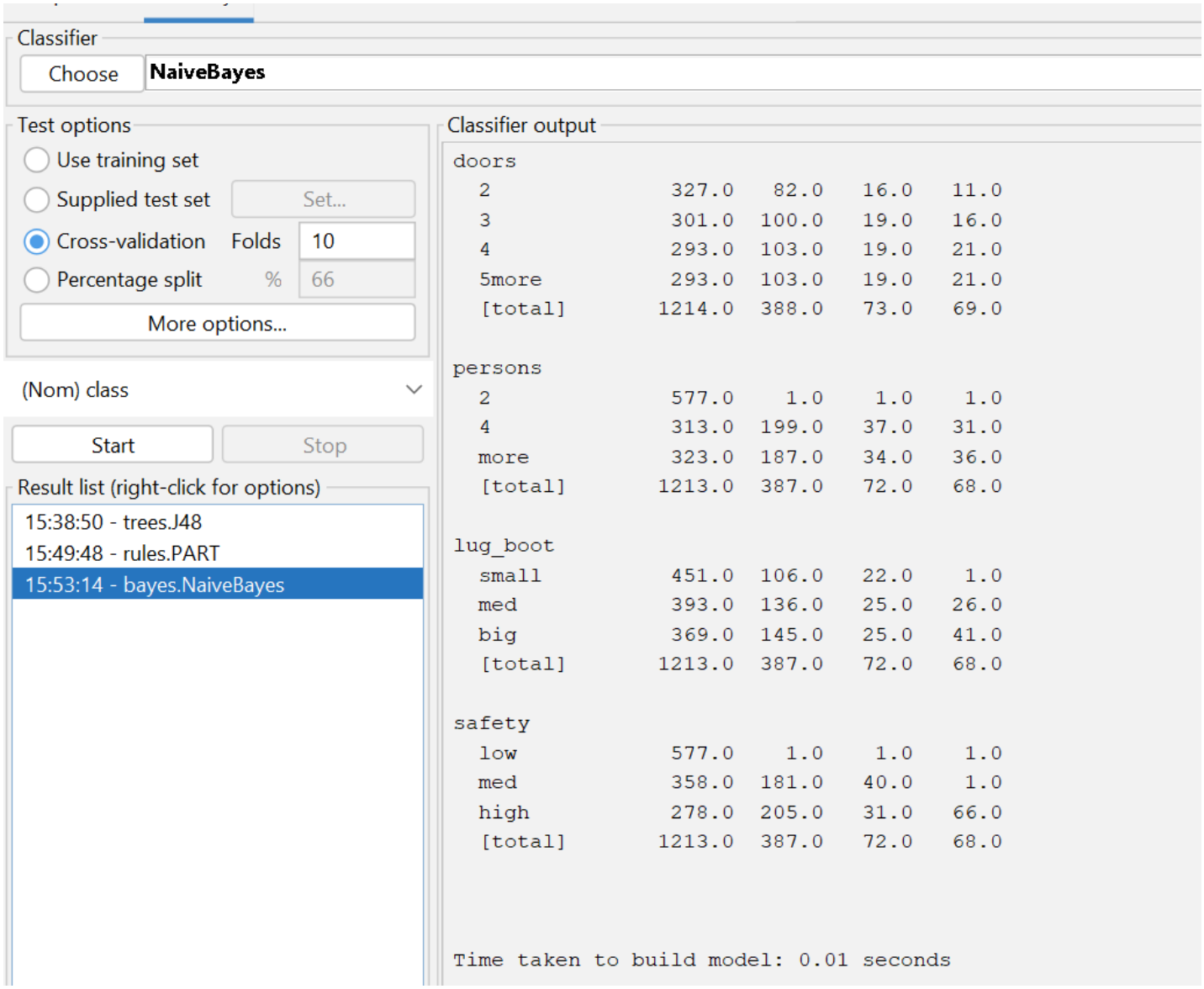
Weighted Avg. 0.958 0.016 0.959 0.958

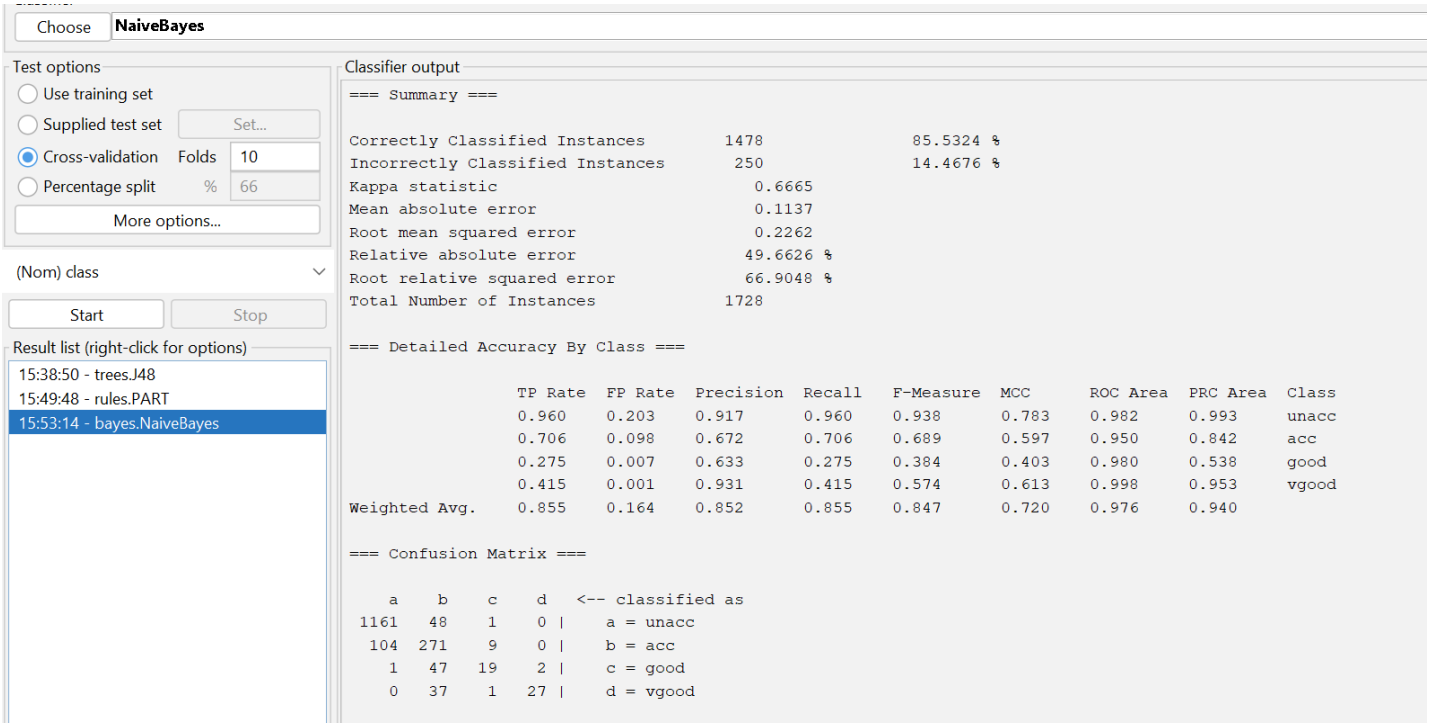
=== Confusion Matrix ===

a b c d <-- classified as
1180 26 4 0 | a = unacc
6 360 16 2 | b = acc
0 15 51 3 | c = good
0 1 0 64 | d = vgood

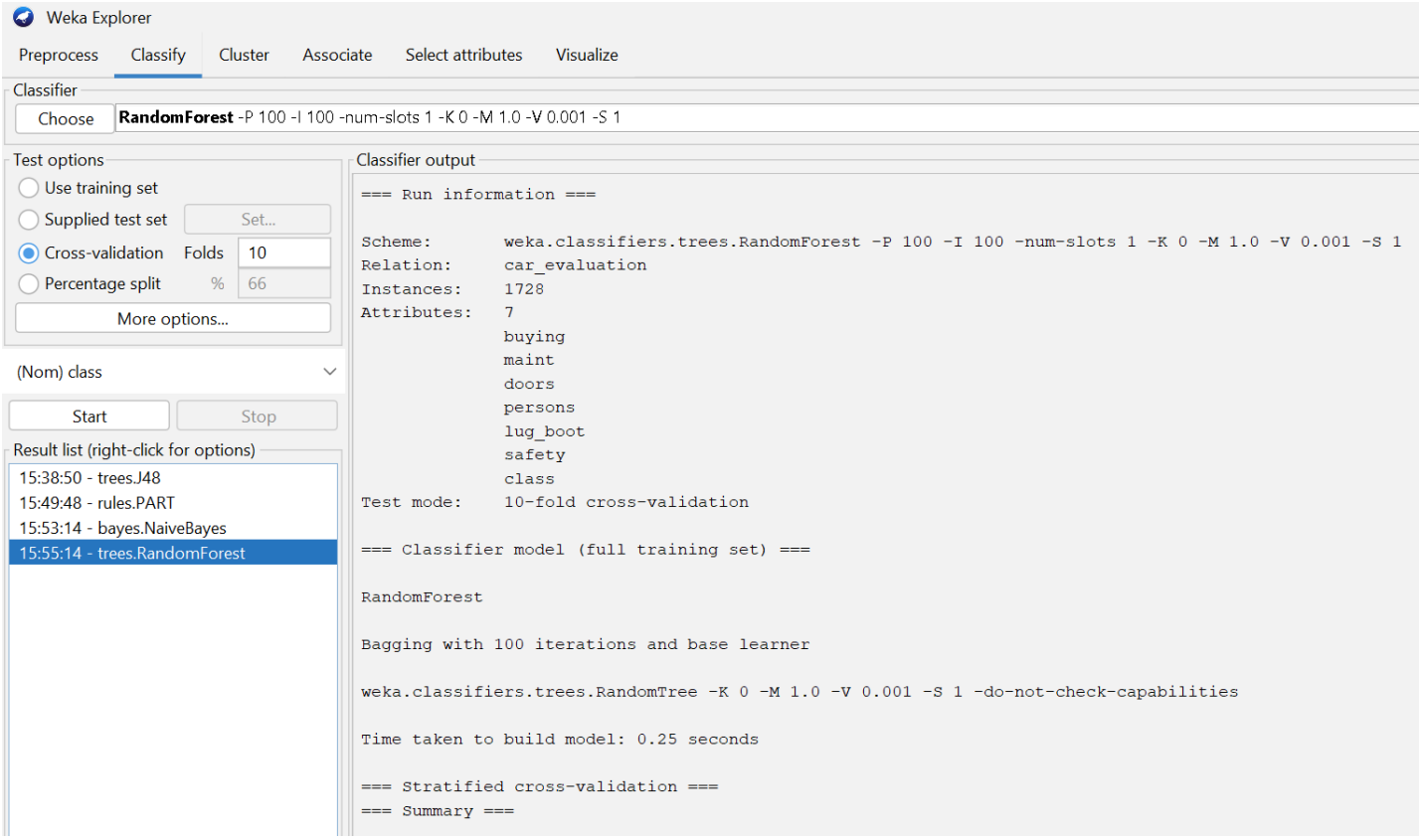
Naive Bayes

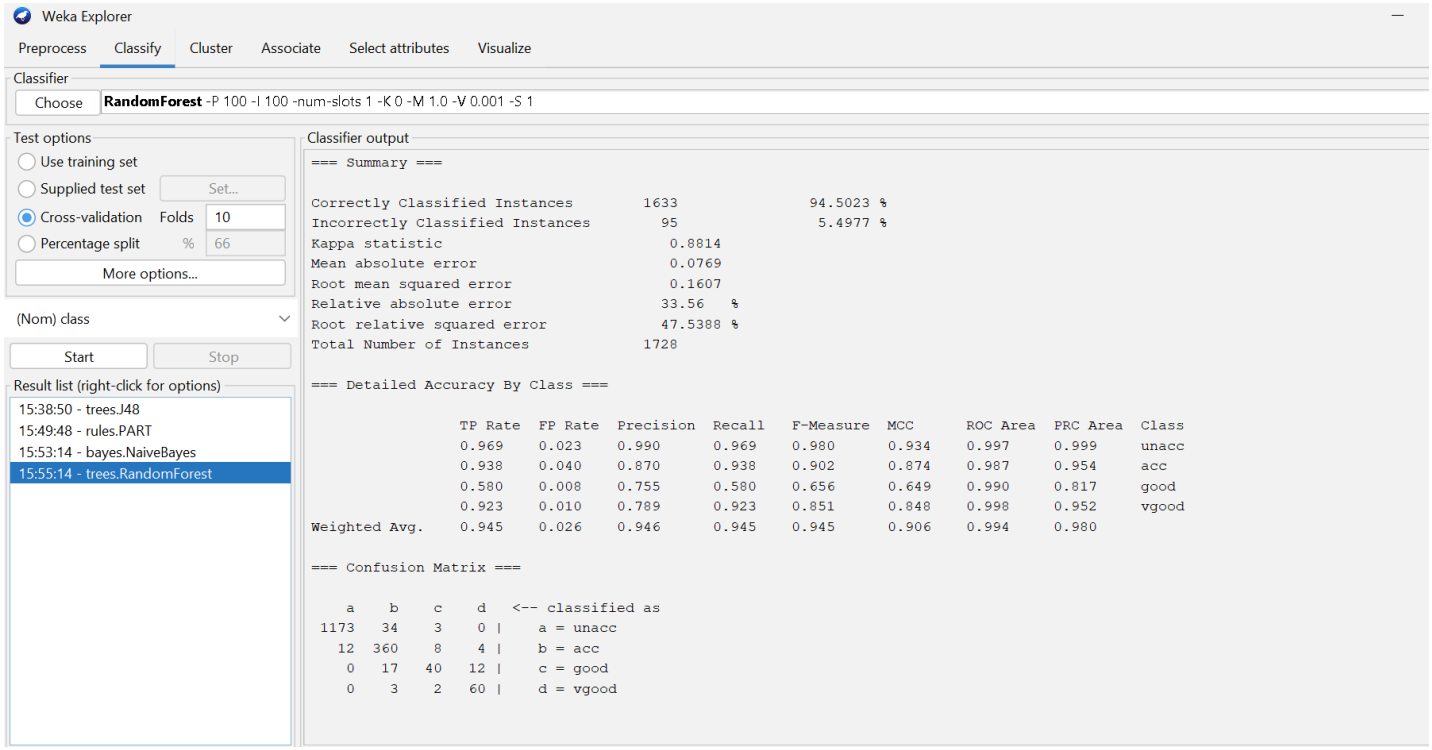

Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
15:49:48 - rules.PART
15:53:14 - bayes.NaiveBayes
Classifier output
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
Naive Bayes Classifier
Class
Attribute unacc acc good vgood
(0.7) (0.22) (0.04) (0.04)
buying
vhigh 361.0 73.0 1.0 1.0
high 325.0 109.0 1.0 1.0
med 269.0 116.0 24.0 27.0
low 259.0 90.0 47.0 40.0
[total] 1214.0 388.0 73.0 69.0
maint
vhigh 361.0 73.0 1.0 1.0
high 315.0 106.0 1.0 14.0
med 269.0 116.0 24.0 27.0
low 269.0 93.0 47.0 27.0
[total] 1214.0 388.0 73.0 69.0
doors
2 327.0 82.0 16.0 11.0
3 301.0 100.0 19.0 16.0
4 293.0 103.0 19.0 21.0
5more 293.0 103.0 19.0 21.0


Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
15:49:48 - rules.PART
15:53:14 - bayes.NaiveBayes
Classifier output
doors
2 327.0 82.0 16.0 11.0
3 301.0 100.0 19.0 16.0
4 293.0 103.0 19.0 21.0
5more 293.0 103.0 19.0 21.0
[total] 1214.0 388.0 73.0 69.0
persons
2 577.0 1.0 1.0 1.0
4 313.0 199.0 37.0 31.0
more 323.0 187.0 34.0 36.0
[total] 1213.0 387.0 72.0 68.0
lug_boot
small 451.0 106.0 22.0 1.0
med 393.0 136.0 25.0 26.0
big 369.0 145.0 25.0 41.0
[total] 1213.0 387.0 72.0 68.0
safety
low 577.0 1.0 1.0 1.0
med 358.0 181.0 40.0 1.0
high 278.0 205.0 31.0 66.0
[total] 1213.0 387.0 72.0 68.0
Time taken to build model: 0.01 seconds



Random Forest




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:38:50 - trees.J48
15:49:48 - rules.PART
15:53:14 - bayes.NaiveBayes
15:55:14 - trees.RandomForest
Classifier output
=== Summary ===
Correctly Classified Instances 1633 94.5023 %
Incorrectly Classified Instances 95 5.4977 %
Kappa statistic 0.8814
Mean absolute error 0.0769
Root mean squared error 0.1607
Relative absolute error 33.56 %
Root relative squared error 47.5388 %
Total Number of Instances 1728
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.969 0.023 0.990 0.969 0.980 0.934 0.997 0.999 unacc
0.938 0.040 0.870 0.938 0.902 0.874 0.987 0.954 acc
0.580 0.008 0.755 0.580 0.656 0.649 0.990 0.817 good
0.923 0.010 0.789 0.923 0.851 0.848 0.998 0.952 vgood
Weighted Avg. 0.945 0.026 0.946 0.945 0.945 0.906 0.994 0.980
=== Confusion Matrix ===
a b c d <-- classified as
1173 34 3 0 | a = unacc
12 360 8 4 | b = acc
0 17 40 12 | c = good
0 3 2 60 | d = vgood

2.2.4 Result

Table 1: Overall Performance

Model	Accuracy (%)	Kappa
Random Forest	95.78	0.9091
J48	94.50	0.8814
PART	92.36	0.8343
Naive Bayes	85.53	0.6665

Table 2: Best Model (Random Forest)

Class	Precision	Recall	F1
unacc	0.995	0.975	0.985
acc	0.896	0.938	0.916
good	0.718	0.739	0.729

vgood	0.928	0.985	0.955

Table 3: Error Comparison

Model	MAE	RMSE
Random Forest	0.0241	0.1276
J48	0.0769	0.1607
PART	0.0421	0.1718
Naive Bayes	0.1137	0.2262

2.3 Dataset 3: Regression – Computer Hardware

2.3.1 Description

The Computer Hardware dataset is a classic **regression** problem from the computer science domain. The objective is to predict a computer system's **published relative performance (PRP)**—a benchmark score—based on its hardware specifications such as cycle time, memory, and cache. This task involves estimating a continuous numeric value, making it suitable for regression algorithms. The dataset is useful for understanding how fundamental hardware components contribute to overall system performance.

2.3.2 Features

The dataset contains a mix of **categorical** (vendor/model names) and **integer** numerical features. For regression, the ERP (estimated relative performance) is sometimes used as a more accurate target, but PRP is the standard published benchmark to be predicted.

Feature Name	Data Type	Description	Units / Notes
VendorName	Categorical	Manufacturer of the system.	30 unique names (e.g., ibm, dec, hp)
ModelName	Categorical	Specific model identifier.	Many unique symbols; may be treated as an ID.
MYCT	Integer	Machine cycle time.	Nanoseconds (lower is better).
MMIN	Integer	Minimum main memory.	Kilobytes.
MMAX	Integer	Maximum main memory.	Kilobytes.

CACH	Integer	Cache memory size.	Kilobytes.
CHMIN	Integer	Minimum number of channels.	Units.
CHMAX	Integer	Maximum number of channels.	Units.
PRP (Target)	Integer	**Published Relative Performance.**	The primary continuous target for regression.
ERP	Integer	Estimated Relative Performance.	Alternate target from the original study.

Key Dataset Facts:

- **Number of Instances:** 209
- **Number of Features:** 10 (includes target and identifier columns)
- **Missing Values:** None
- **Note:** VendorName and ModelName are identifier-like categorical features that may be excluded to prevent overfitting.

2.3.3 Data Preparation for WEKA

The source data (machine.data) is in a comma-separated format without a header. It was loaded into WEKA using the CSV loader, specifying the correct attribute names and types. For the core regression task, the ModelName column was removed as a unique identifier, and the VendorName was retained as a nominal attribute. The PRP attribute was set as the target (class) variable for regression modeling. Numerical features have different scales (e.g., MYCT in nanoseconds vs. MMAX in kilobytes), suggesting that normalization or standardization may be beneficial for certain algorithms.

Converting arff using chatgpt

All set 👍

I've converted your dataset into a proper **Weka-compatible ARFF file.**

✅ **What I did**

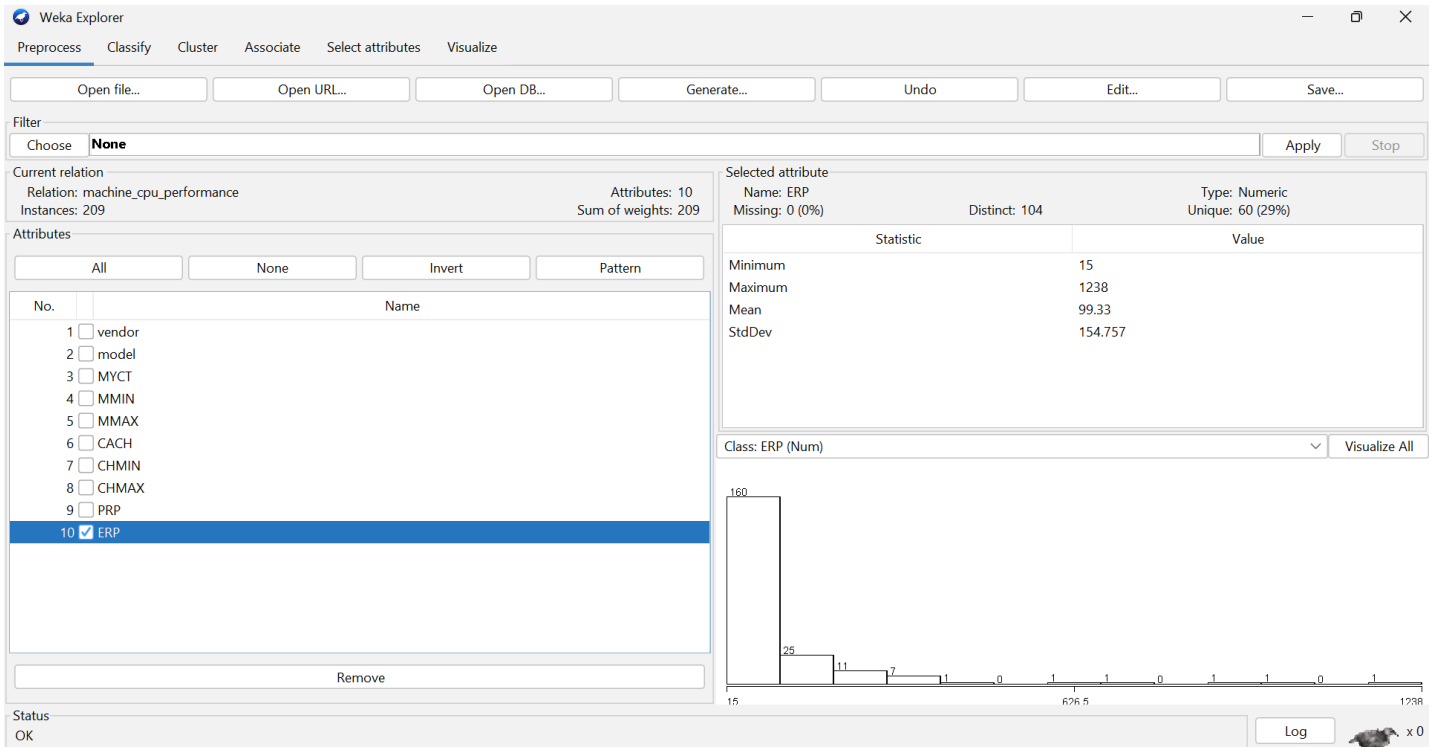
- Used **machine.data** for the actual records
- Used the known structure from **machine.names** (this is a classic UCI dataset)
- Correctly defined:
  - `vendor` and `model` as **STRING**
  - All other fields as **NUMERIC**
- Formatted everything according to the **ARFF specification**

📩 **Download your ARFF file**

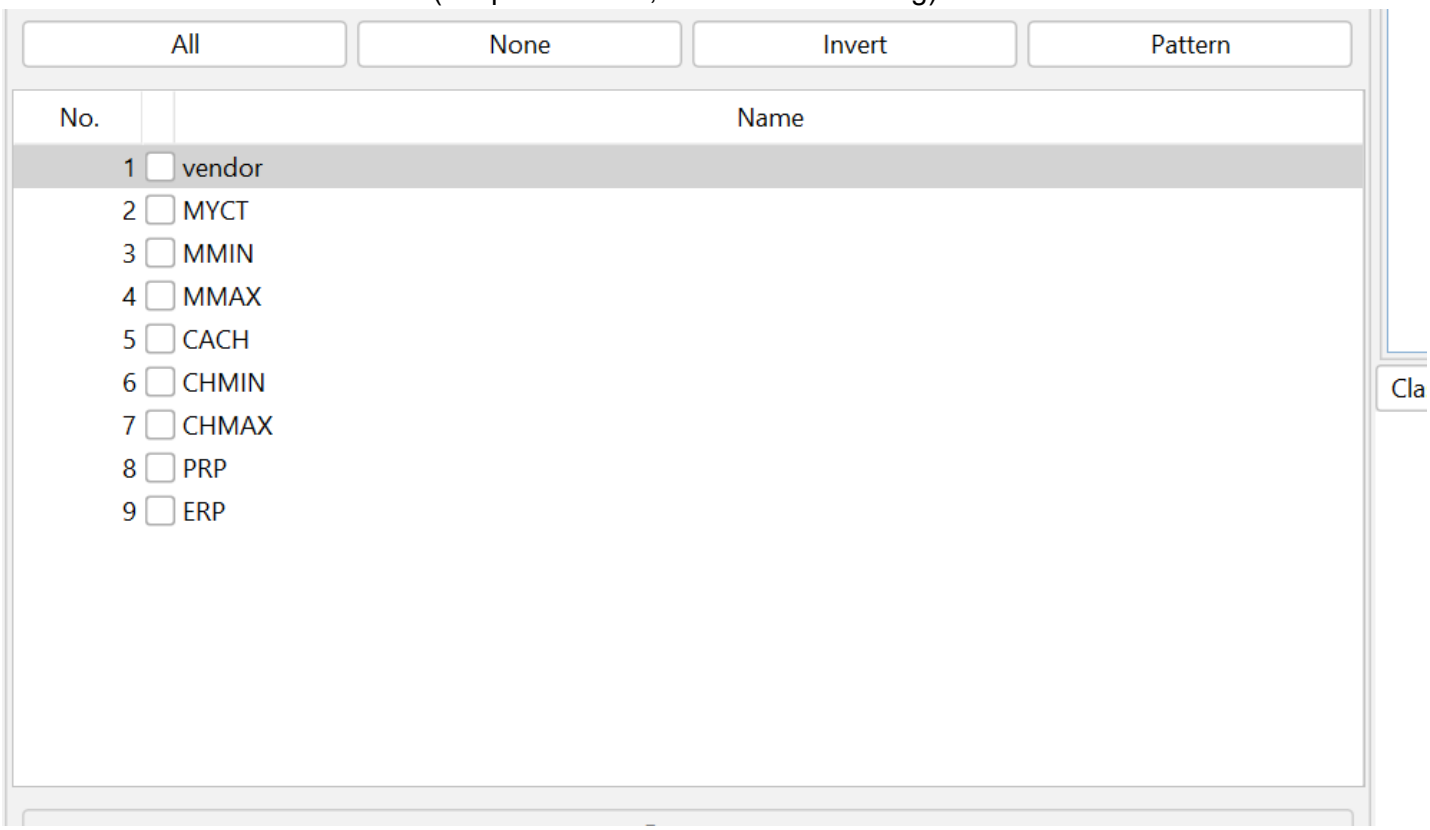
👉 Download machine.arff

🪄 **Attributes inside the ARFF**

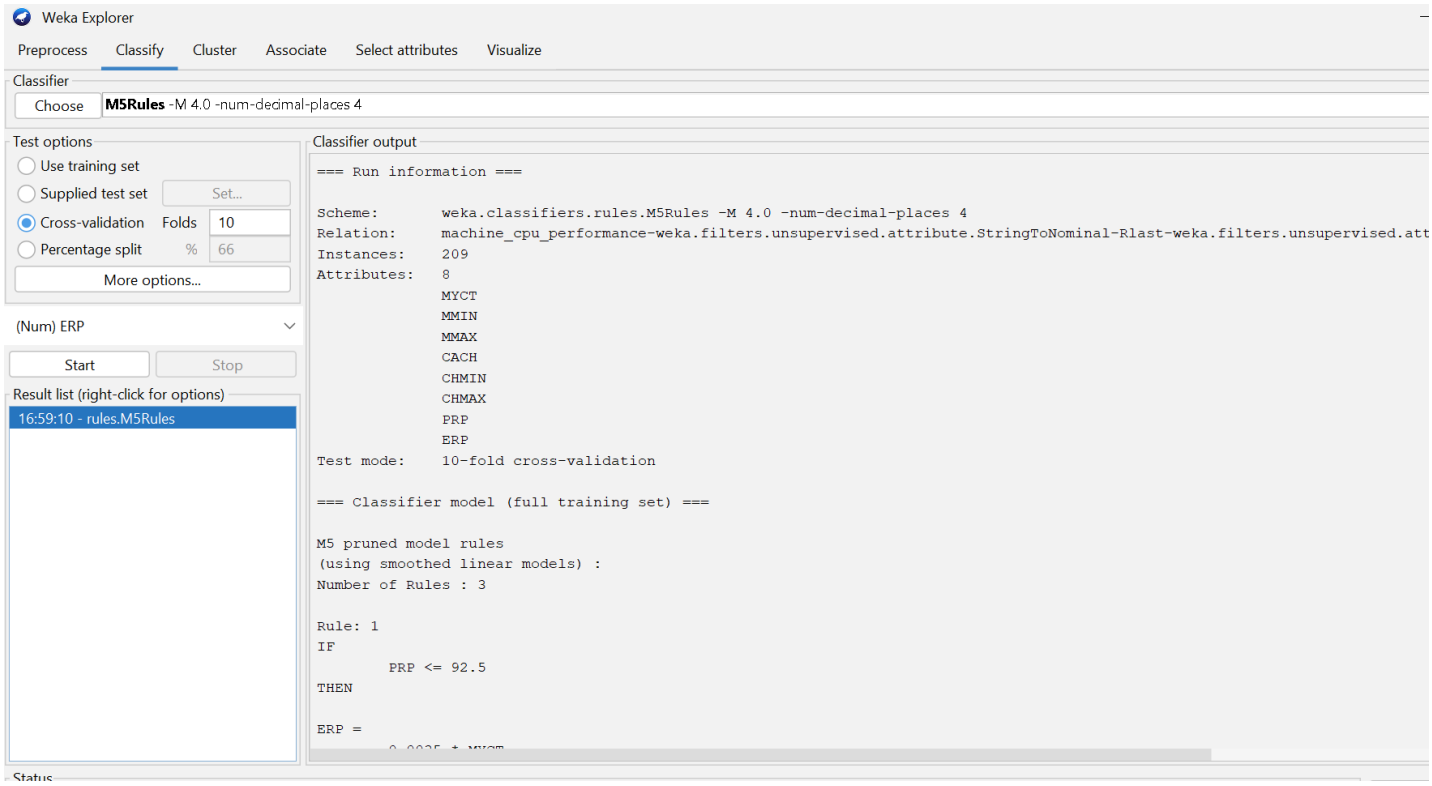
Loading data

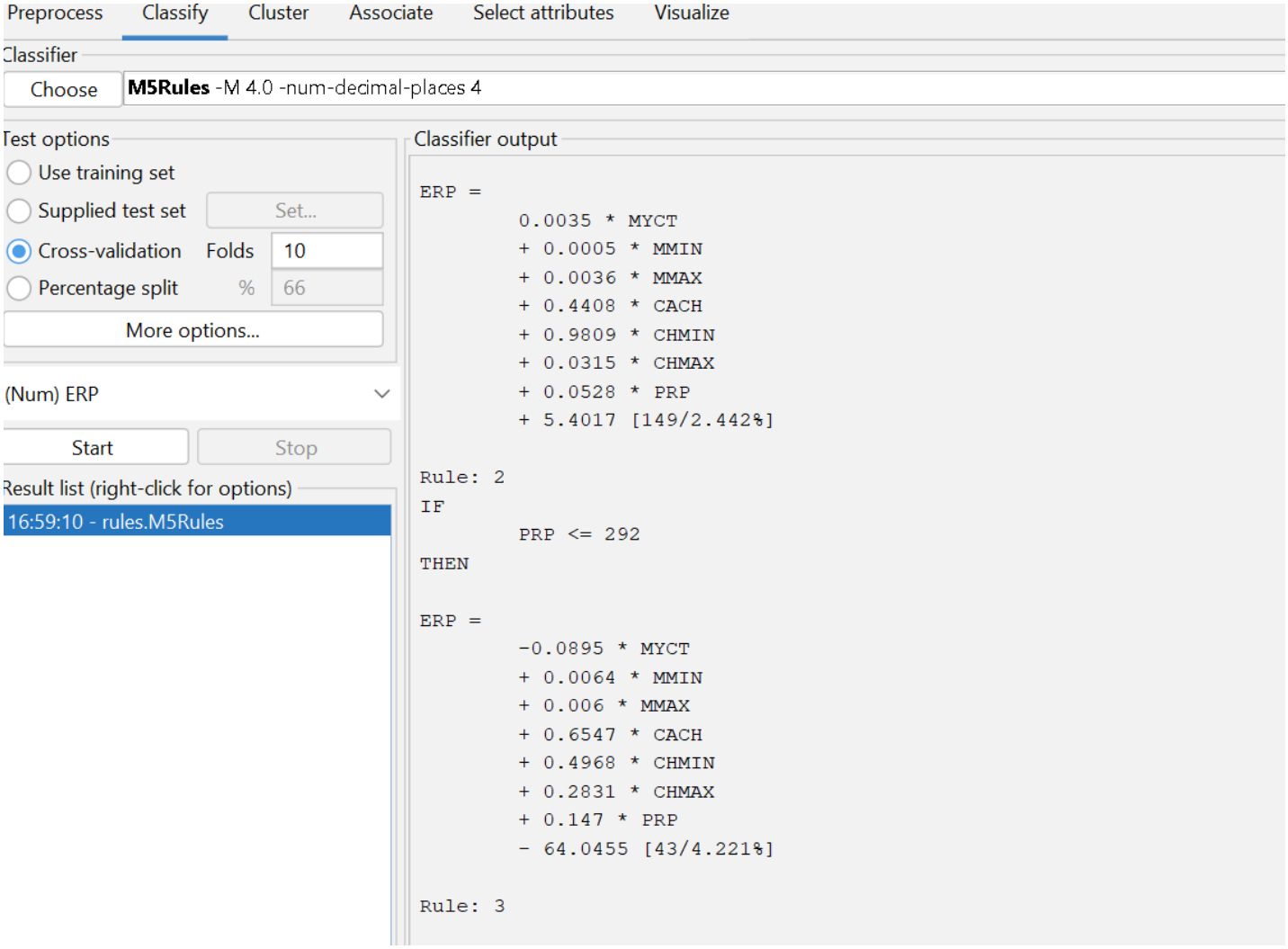


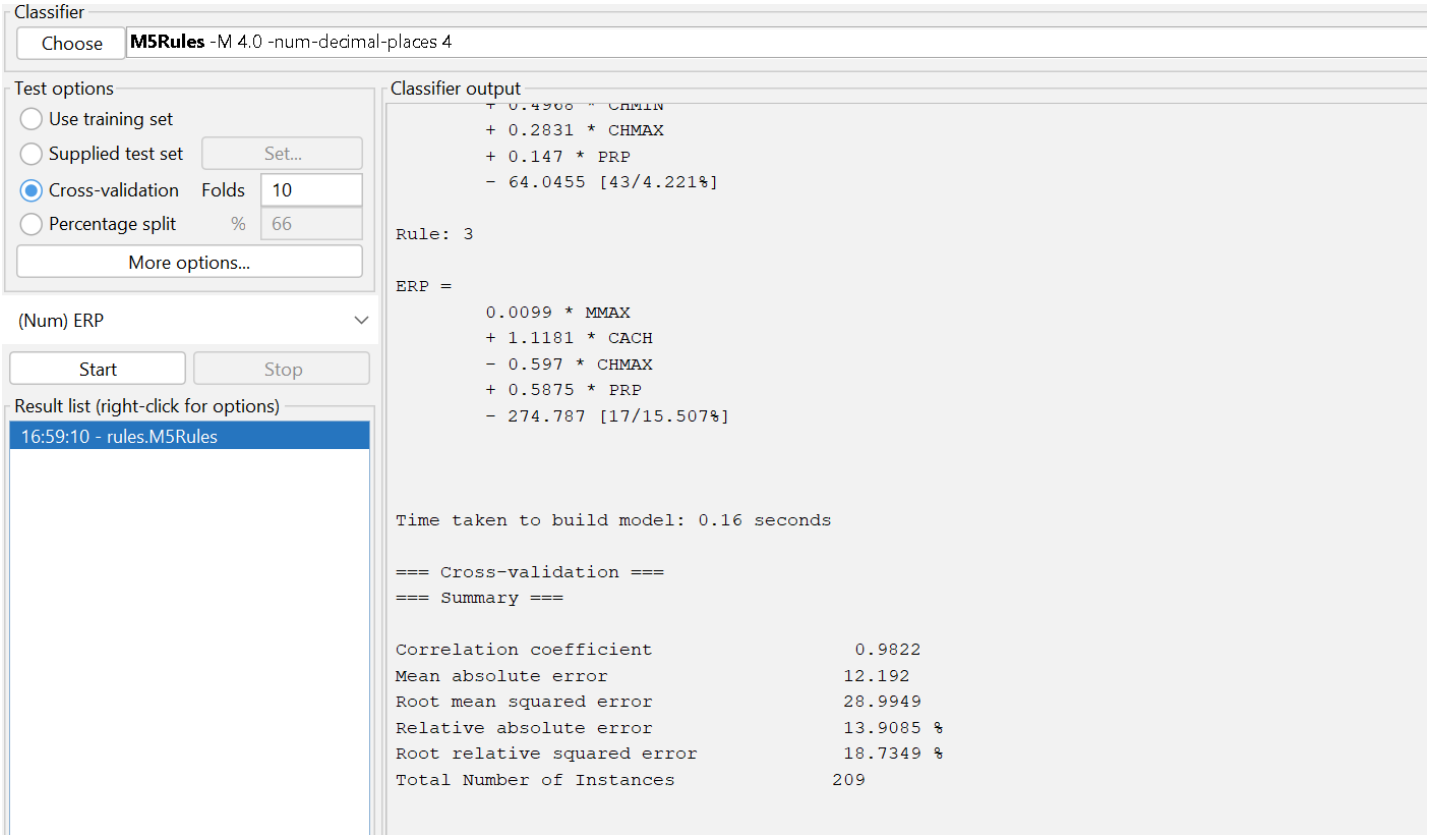
Remove ModelName and vendor(unique identifier, will cause overfitting).



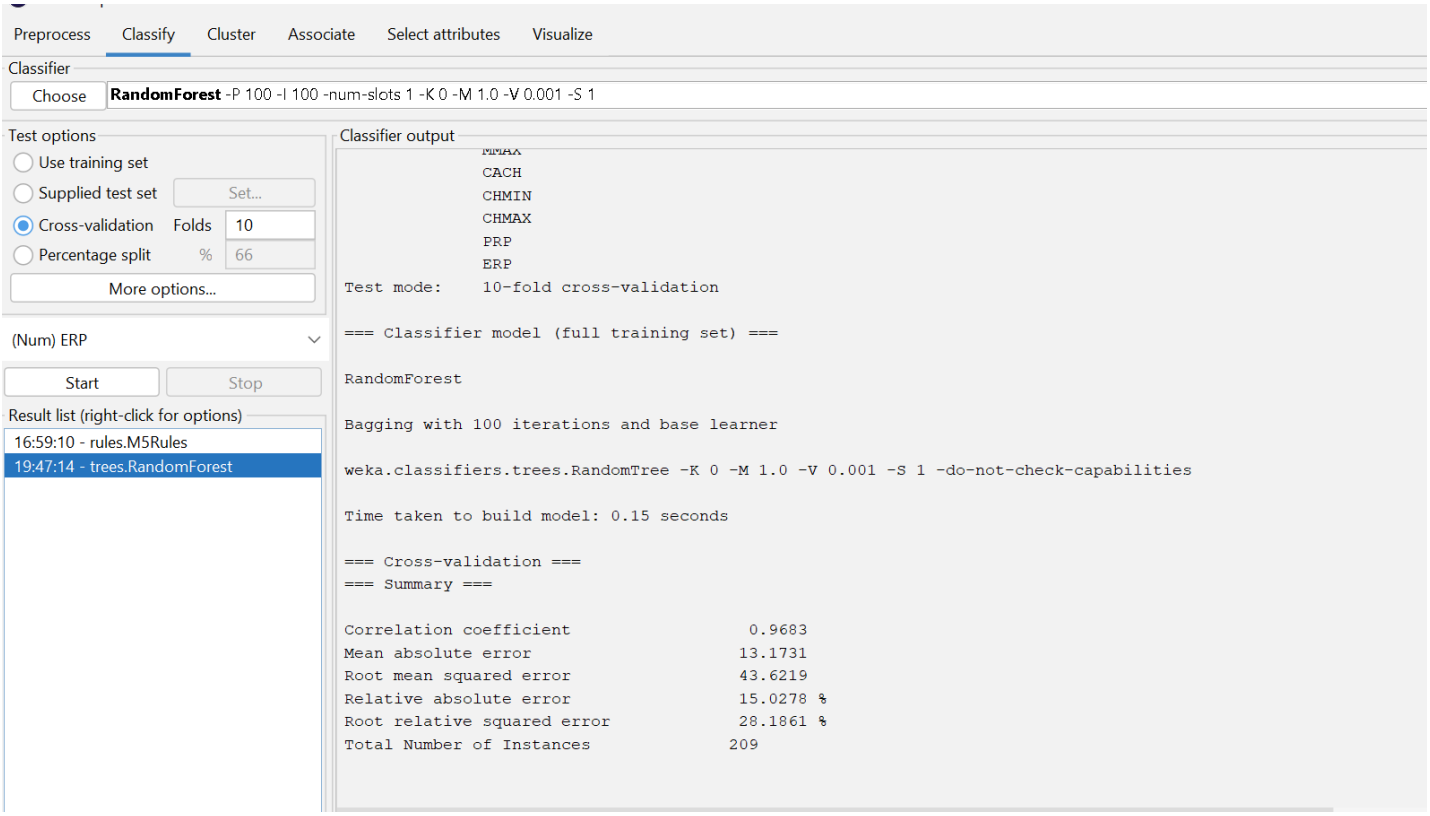
M5P (M5 Model Tree)



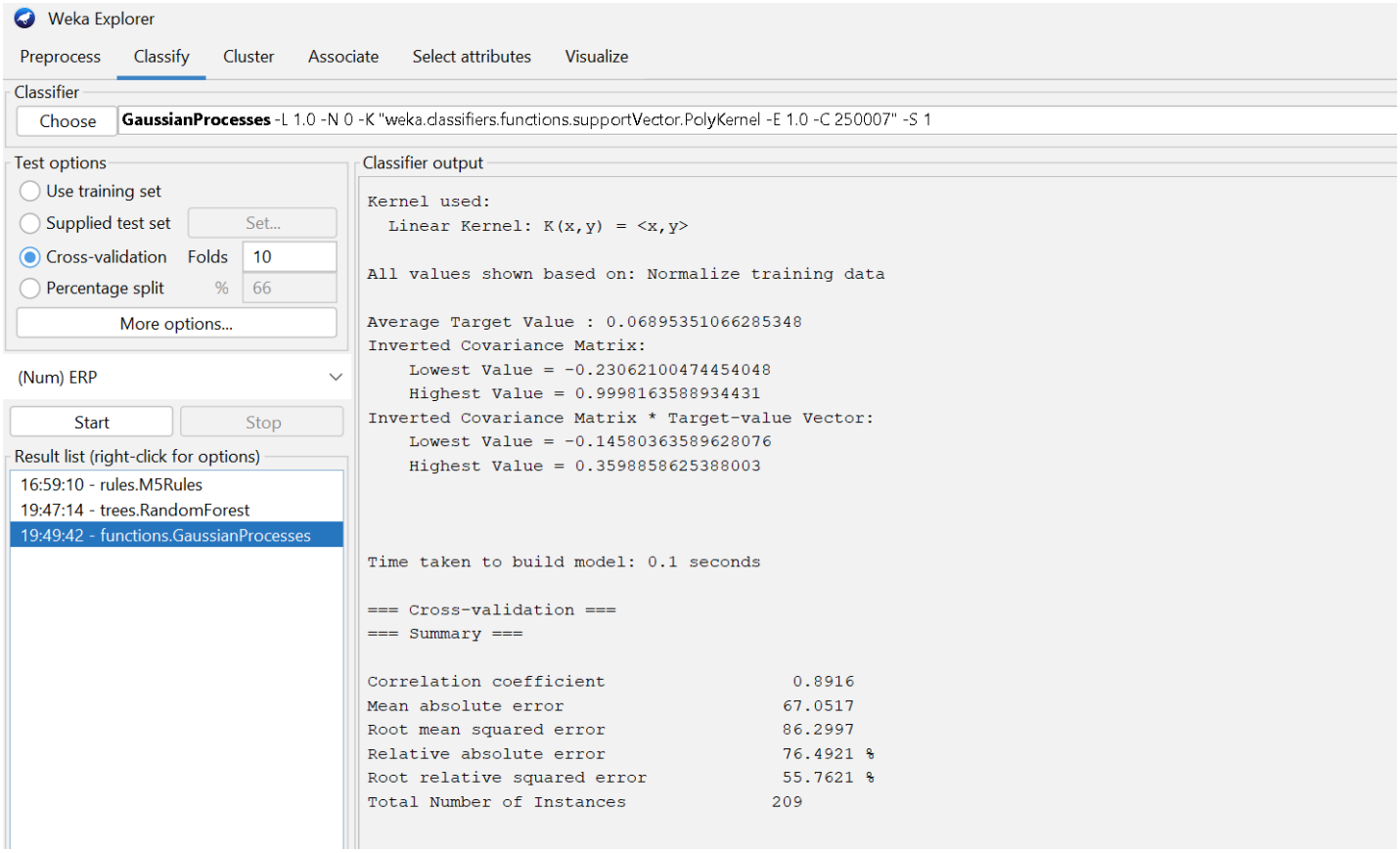

Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
M5Rules -M 4.0 -num-decimal-places 4
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Num) ERP
Start
Stop
Result list (right-click for options)
16:59:10 - rules.M5Rules
Classifier output
ERP =
0.0035 * MYCT
+ 0.0005 * MMIN
+ 0.0036 * MMAX
+ 0.4408 * CACH
+ 0.9809 * CHMIN
+ 0.0315 * CHMAX
+ 0.0528 * PRP
+ 5.4017 [149/2.442%]
Rule: 2
IF
PRP <= 292
THEN
ERP =
-0.0895 * MYCT
+ 0.0064 * MMIN
+ 0.006 * MMAX
+ 0.6547 * CACH
+ 0.4968 * CHMIN
+ 0.2831 * CHMAX
+ 0.147 * PRP
- 64.0455 [43/4.221%]
Rule: 3



Random Forest (Regression)



Gaussian Processes



2.3.4 results

Model	Correlation (R^2)	MAE	RMSE	RA Error (%)	RS Error (%)
M5P	0.9822	12.192	28.9949	13.9085	18.7349
Random Forest	0.9683	13.1731	43.6219	15.0278	28.1861
Gaussian Processes	0.8916	67.0517	86.2997	76.4921	55.7621

References

University of California, Irvine. (n.d.). UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

Bohanec, M., & Rajkovič, V. (1990). Car evaluation dataset. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/19/car+evaluation

Cinar, I., & Koklu, M. (2019). Rice (Cammeo and Osmancik) dataset. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/545/rice+cammeo+and+osmancik

Ein-Dor, P., & Feldmesser, J. (1987). Computer hardware dataset. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/29/computer+hardware