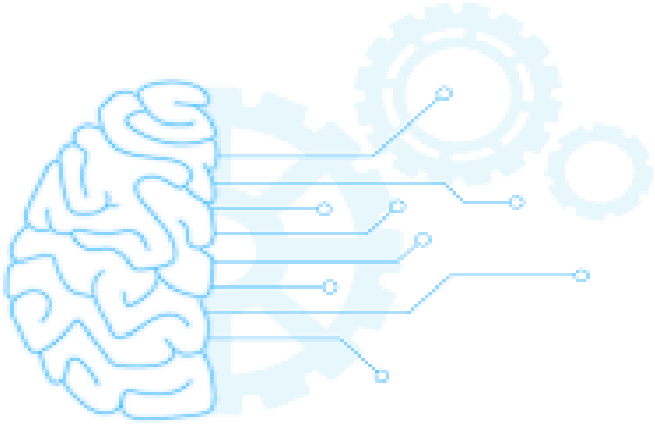
Machine Learning



Room Occupancy Estimation Dataset

Artificial Intelligence (CPCS 331)

Team Members:

Hanin Suleiman Al-haj – 2010269

Alaa Emad Al-hamzi – 2010304

Roaa Abdullah Al-zahrazi – 2005863

Najd Khalid Al-otaibi – 2006156

Section: B**3**A

Hand in date: 6-11-2022

Task Assignment

Each member of the team has selected one ML algorithm to apply to the same dataset we chose (Click here to see the data set), the following table shows the algorithm that has been distributed among team members.

Name	Algorithm
Hanin Sulieman	Decision Stump
Alaa Emad	Naïve Bays
Roaa Alzahrani	Random Tree
Najd Alotaibi	Forest Tree

Each member should apply the chosen algorithm on the same dataset in both **Weka** and **RapidMiner** and then present the output and the accuracy of the used algorithm after using the validation tool and changing the validation options multiple times.

Table of Contents

Illustrations:

Tables:

Figures:

1. Introduction

1.1 The Topic of the Report

The dataset that we choose to work on in this report is the room occupancy estimation dataset. In buildings, a large amount of energy is usually spent on heating, ventilation, and air conditioning systems. There are various ways that have been used to optimize their usage. One of these useful ways is to make them demand-driven depending on human occupancy.

Our data set focuses on estimating the precise number of occupants in a room using various environmental sensors, such as temperature, light, sound, CO2, and others.

1.2 Problem Explanation

The problem we aim to solve is finding the most appropriate algorithm among all the algorithms we chose that could build a model on our dataset with the highest accurate percentage possible after examining it by the split validation.

1.3 Project Purpose

This project aims to introduce a brief idea about how to perform machine learning algorithms on a particular dataset. Moreover, calculating the accuracy of the machine learning by separating the data set into training data and other data as testing data.

1.4 Outline the Approach

Firstly, we chose the ROOM OCCUPANCY DATASET which contains the following attributes: temperature, light, sound, CO2, CO2 slope, and PIR, to determine how many people exist in a room 0, 1, 2, and 3.

Secondly, using **Weka** and **RapidMiner,** we applied different ML algorithms on the dataset.

Thirdly, we used cross-validation and split-validation approaches to calculate the accuracy of the chosen algorithm. Also, we compared the accuracy between both programs.

Finally, we analysed the result of 4 algorithms on the same dataset and compared their accuracy.

In my report, I chose the Decision Stump algorithm to test the dataset.

2. Technical Description

2.1 Describe the Data Set

The data set we choose to work on is composed of 10129 instances and 16 attributes, such as temperature, light, sound, CO2, and others. Also, it has one label, which is the (Room Occupancy Count) attribute.

Note: We conducted a modification on the data of the Room Occupancy Count attribute (None, Single, Double, Triple)

The associated task with the data set is classification. The decision that we will get from these data is to estimate the precise number of occupants in a room using various environmental sensors (attributes).

The attributes are:

- Date
- Time
- Temperature
- Light
- Sound
- CO2
- CO2 Slope
- PIR
- Room Occupancy Count

2.2 Describe the ML Algorithm

The algorithm that I chose is the Decision Stump algorithm. The Decision Stump algorithm is a binary classification algorithm. It is one of the most straightforward classification algorithms. It is a Decision Tree, that uses only a single attribute for splitting. The idea of this algorithm is that each time we have to focus on only one feature and find a point that can separate data the most.

3. Result

3.1 Experiment Results

- **Experiment Result of my algorithm (Decision Stump) in both RapidMiner and Weka:**

Percentages	RapidMiner Accuracy	Weka Accuracy
70% training data, 30% test data	88.61%	88.8779%
80% training data, 20% test data	88.60%	88.6476%
Cross validation (10 Folds)	88.61%	88.607%

Table 1: Table of accuracy in both software and both validations using the Decision Stump algorithm

- **Experiment Result of the four algorithms (Decision Stump, Naïve Bays, Random Tree, Forest Tree) in RapidMiner:**

The Algorithm	Cross Validation Accuracy(10 folds)	Split Validation Accuracy 70%	Split Validation Accuracy 80%
Naïve Bayes	95.83%	95.27%	95.50%
Random Tree	89.90%	87.62%	87.52%
Decision Stump	88.61%	88.61%	88.60%
Random Forest	99.63%	99.61%	99.65%

Table 2: RapidMiner Results

- **Experiment Result of the four algorithms (Decision Stump, Naïve Bays, Random Tree, Forest Tree) in Weka:**

The Algorithm	Split Validation Accuracy 70%	Split Validation Accuracy 80%	Cross Validation (10 folds)
Naïve Bayes	95.19%%	94.71%	95.10%
Random Tree	91.08%	80.95%	86.68%
Decision Stump	88.8779%	88.6476%	88.607%
Random Forest	98.7825%	98.0257%	98.0452%

Table 3: Weka Results

3.2 Analyze the Results

1- Analyze the Decision Stump (my algorithm):

After observing the experiment results of my algorithm that have been previously shown, we can say that both software -RapidMiner and Weka- and both validations -Cross and Split- give extremely close accuracy values. Both software and both validations give accuracy values that range between 88.60% to 88.8779%.

However, it is a better option to implement the Decision Stump algorithm using the Weka software because it has more accurate results.

2- Analyze the four algorithms (Naïve Bayes, Random Tree, Decision Stump, and Random Forest):

Based on the illustrated results of four algorithms we conclude that the most appropriate algorithm for our dataset is the Random Forest algorithm. It has the highest accuracy among other algorithms in both programs ***RapidMiner*** and ***Weka*** and in both split validation and cross-validation.

In the Random Forest algorithm, the RapidMiner software gives higher accuracy than the Weka software. Moreover, the accuracy of the split validation on RapidMiner software has higher accuracy than the cross-validation.

- The Random Forest algorithm accuracy ranges between 98.02% to 99.65%.
- The Naïve Bayes algorithm accuracy ranges between 94.71% to 95.50%.
- The Decision Stump algorithm accuracy ranges between 88.60% to 88.8779%.
- The Random Tree algorithm accuracy ranges between 80.95% to 91.08%.

4. Conclusion

Finally, after analyzing and comparing the results of the algorithms of team members, we concluded that the best-applied algorithm on the room occupancy dataset is the Random Forest algorithm. Which provides the highest accuracy values among other algorithms. Its accuracy reached 99.65%.

5. References

- Decision Stump. (2011). SpringerLink. https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_202?error=cookies_not_supported&code=8e48da85-a73c-4c55-bf06-5de3922c93cd

- Hsu, R. (2022, May 2). Decision Stump - Geek Culture. Medium. https://medium.com/geekculture/decision-stump-b8e93c1f54d7

6. Appendices

6.1 RapidMiner Software

The Cross Validation:

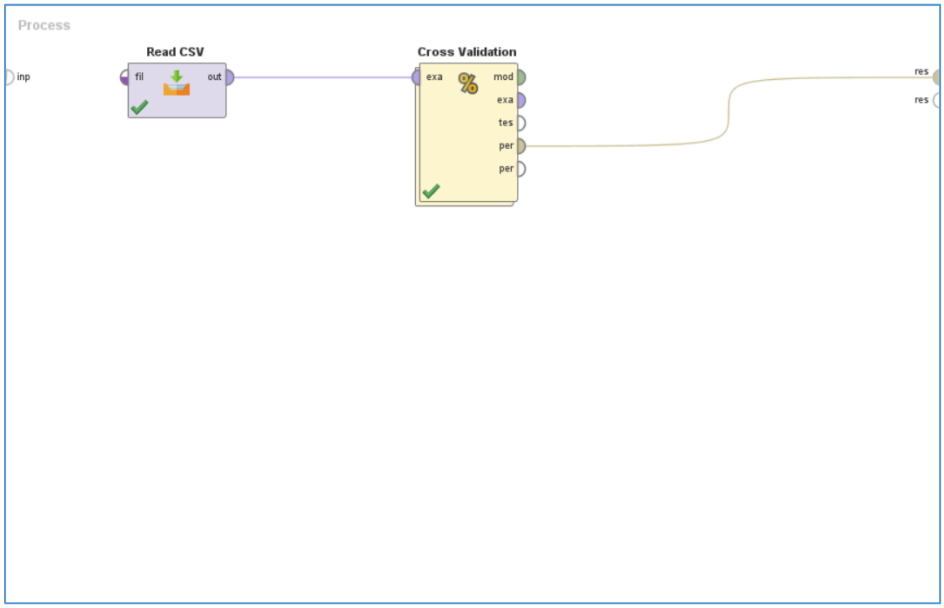


Figure 1: A screenshot of RapidMiner (cross validation)

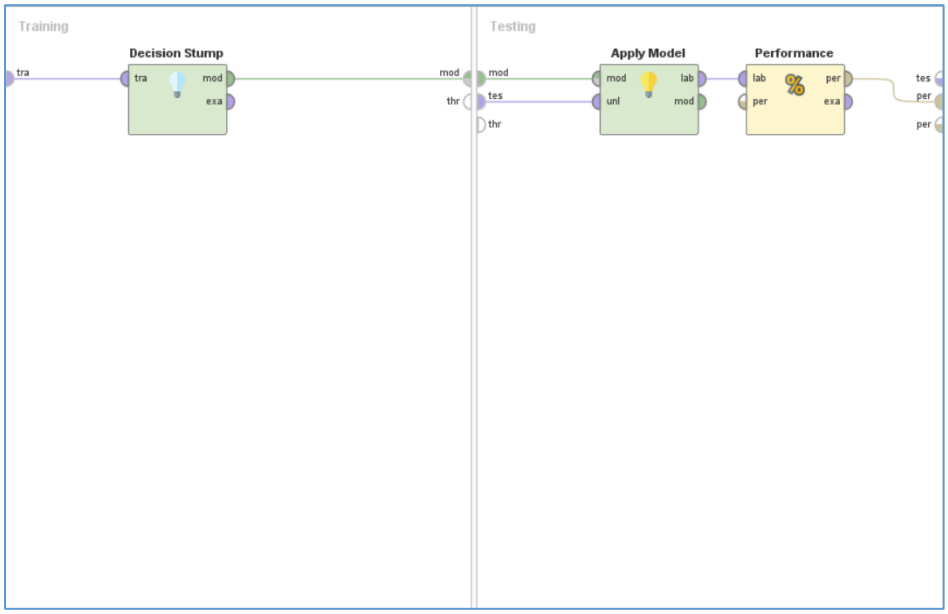


Figure 2: A screenshot of cross validation tool with 10 folds.

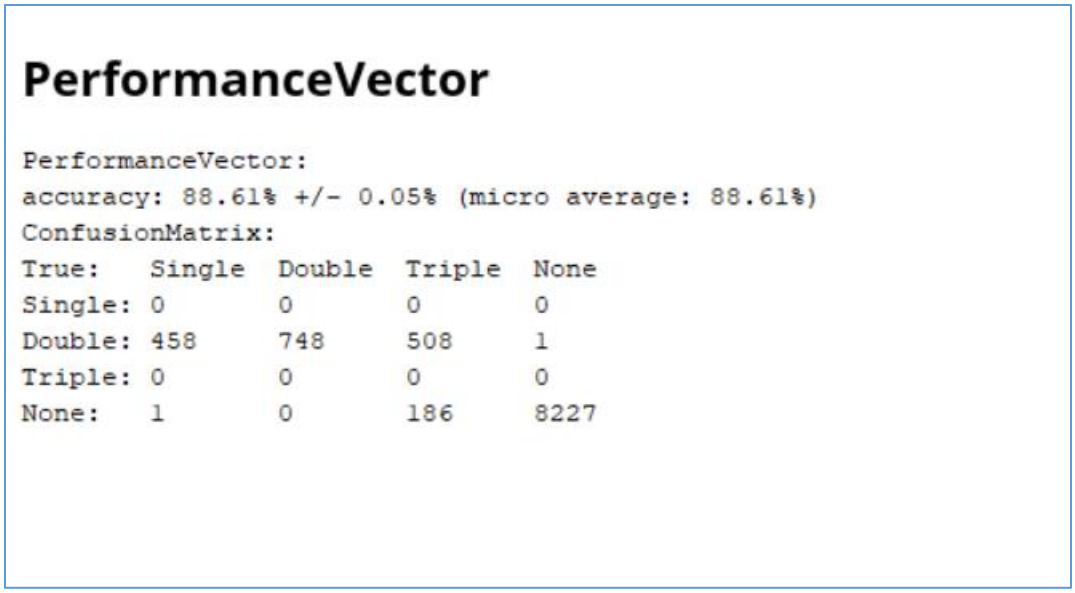
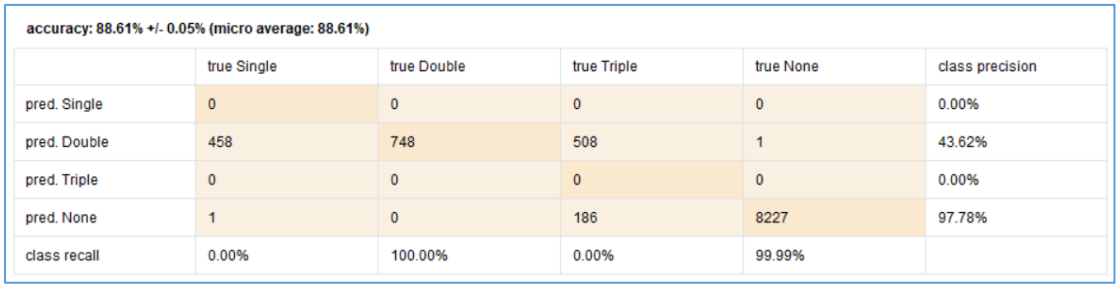


Figure 3: The performance description of RapidMiner (cross validation).



accuracy: 88.61% +/- 0.05% (micro average: 88.61%)

	true Single	true Double	true Triple	true None	class precision
pred. Single	0	0	0	0	0.00%
pred. Double	458	748	508	1	43.62%
pred. Triple	0	0	0	0	0.00%
pred. None	1	0	186	8227	97.78%
class recall	0.00%	100.00%	0.00%	99.99%	

Figure 4: The accuracy of RapidMiner (cross validation).

The Split Validation:

1- The Split Validation with 0.7 Split Ratio:

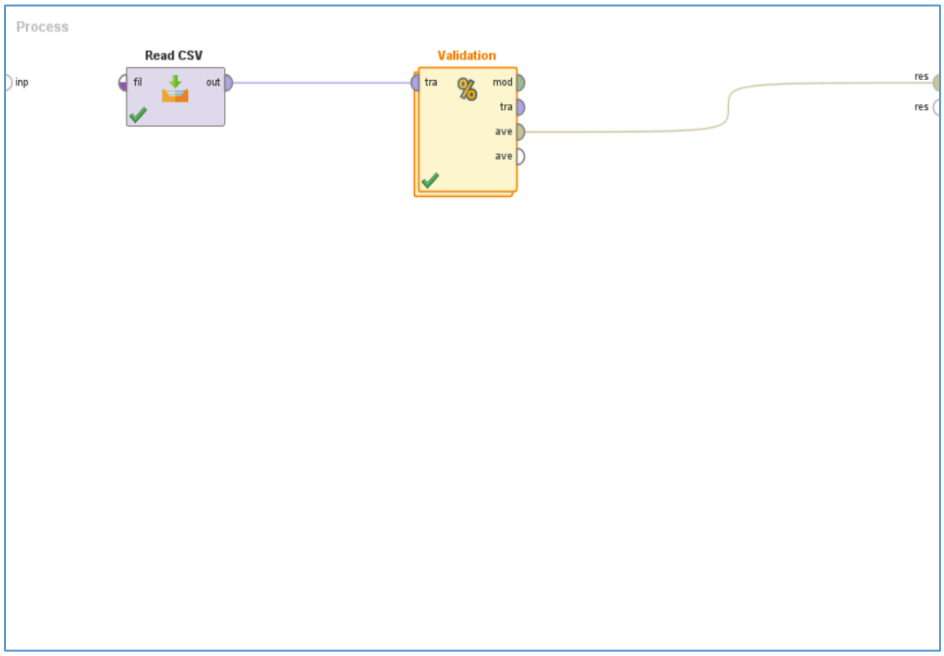


Figure 5: A screenshot of RapidMiner (split validation with 0.7 split ratio)

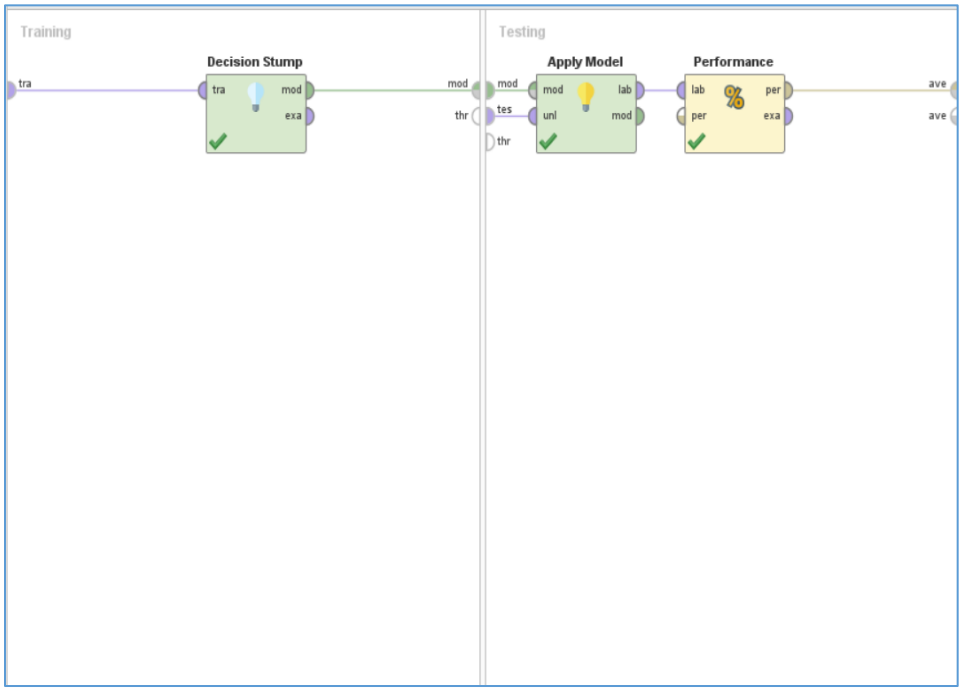


Figure 6: A screenshot of split validation tool with 0.7 split ratio

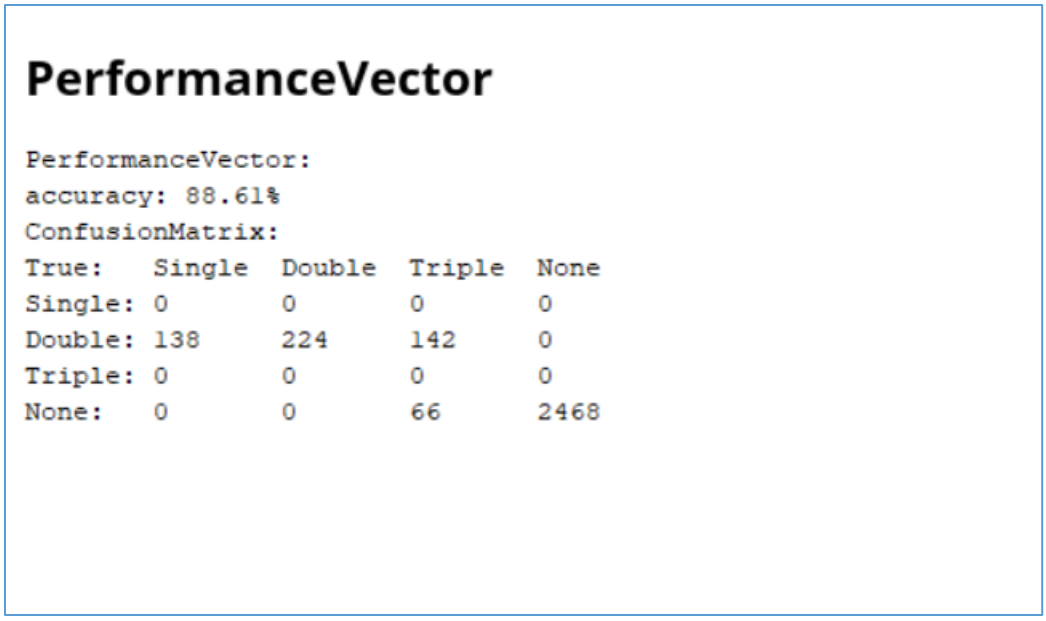
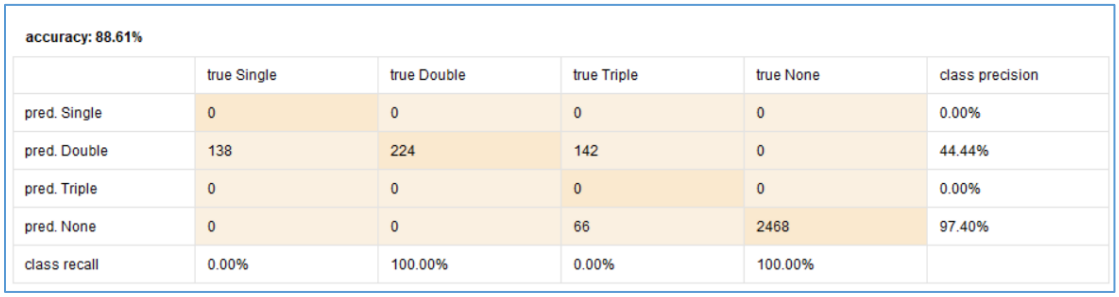


Figure 7: The performance description of RapidMiner (split validation with 0.7 split ratio).



accuracy: 88.61%

	true Single	true Double	true Triple	true None	class precision
pred. Single	0	0	0	0	0.00%
pred. Double	138	224	142	0	44.44%
pred. Triple	0	0	0	0	0.00%
pred. None	0	0	66	2468	97.40%
class recall	0.00%	100.00%	0.00%	100.00%	

Figure 8: The accuracy of RapidMiner (split validation with 0.7 split ratio).

2. The Split Validation with 0.8 Split Ratio:

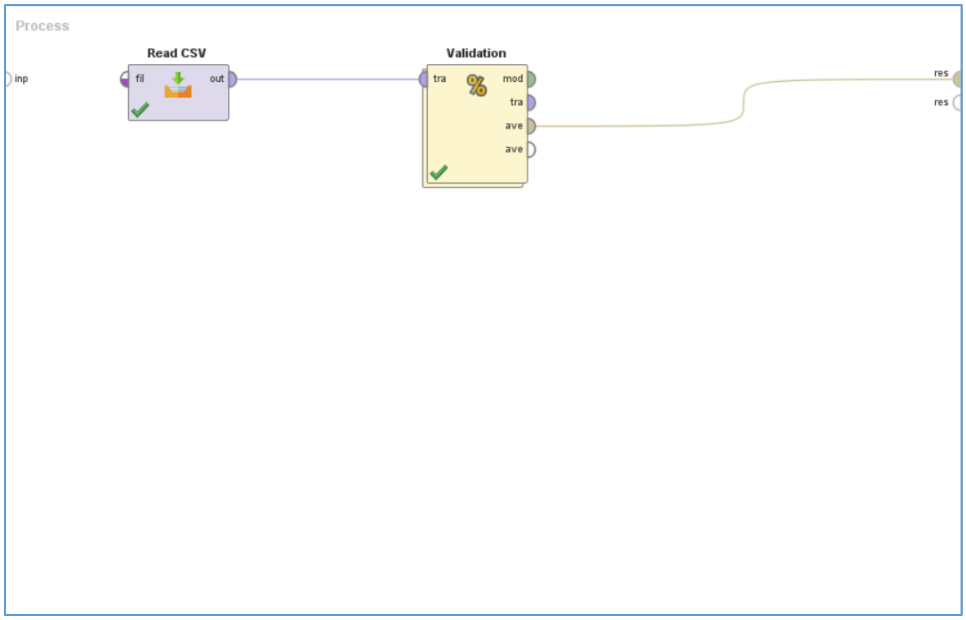


Figure 9: A screenshot of RapidMiner (split validation with 0.8 split ratio)

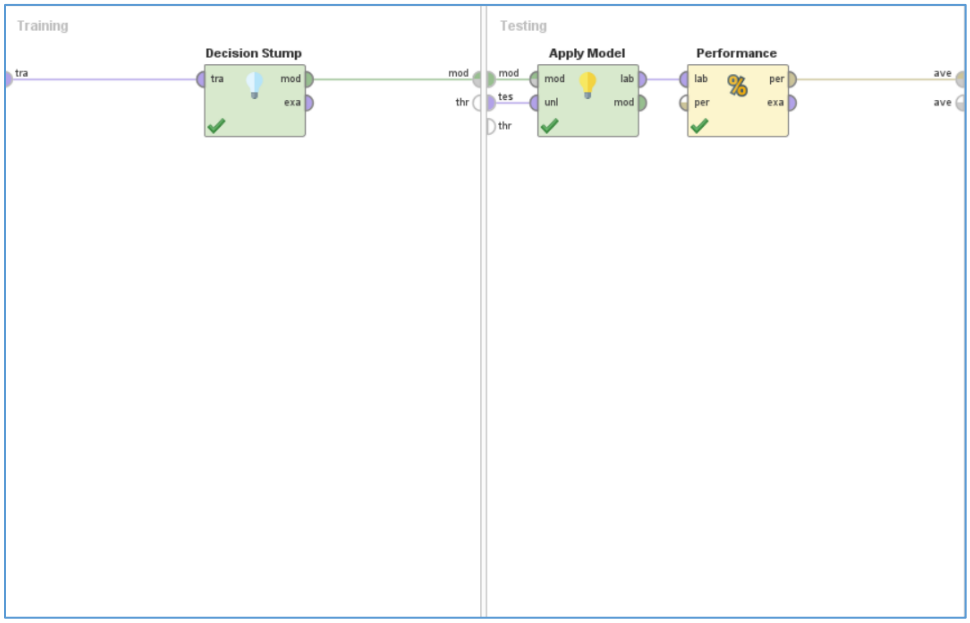


Figure 10: A screenshot of split validation tool with a 0.8 split ratio.

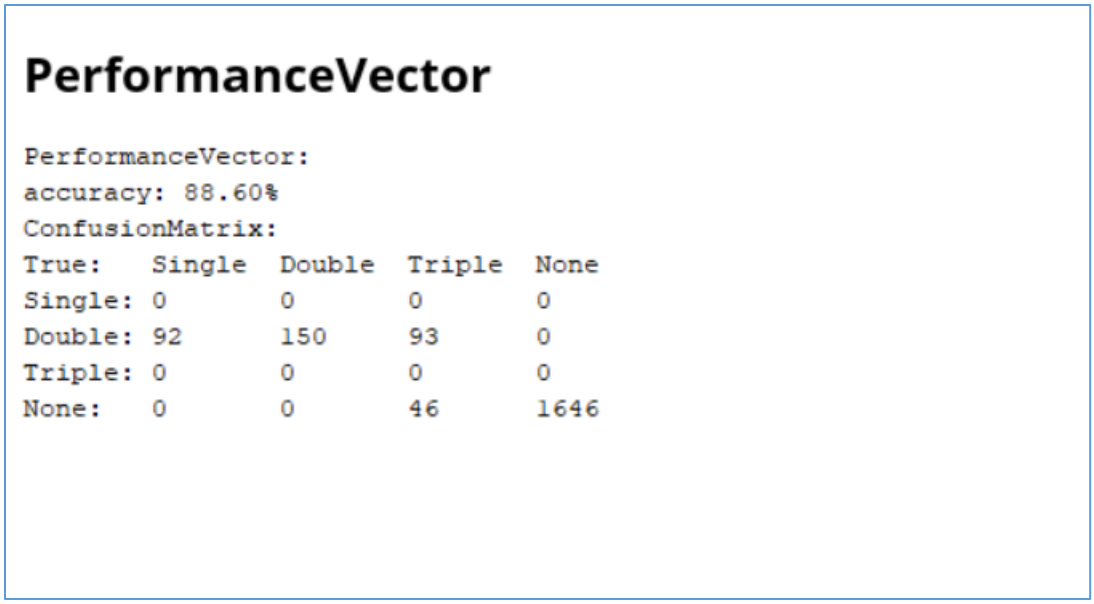
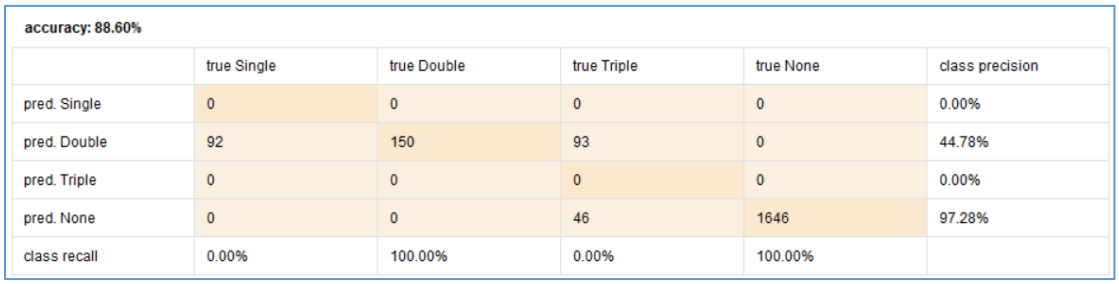


Figure 11: The performance description of RapidMiner (split validation with 0.8 split ratio)



accuracy: 88.60%

	true Single	true Double	true Triple	true None	class precision
pred. Single	0	0	0	0	0.00%
pred. Double	92	150	93	0	44.78%
pred. Triple	0	0	0	0	0.00%
pred. None	0	0	46	1646	97.28%
class recall	0.00%	100.00%	0.00%	100.00%	

Figure 12: The accuracy of RapidMiner (split validation with 0.8 split ratio)

6.2 Weka Software

The Cross Validation:

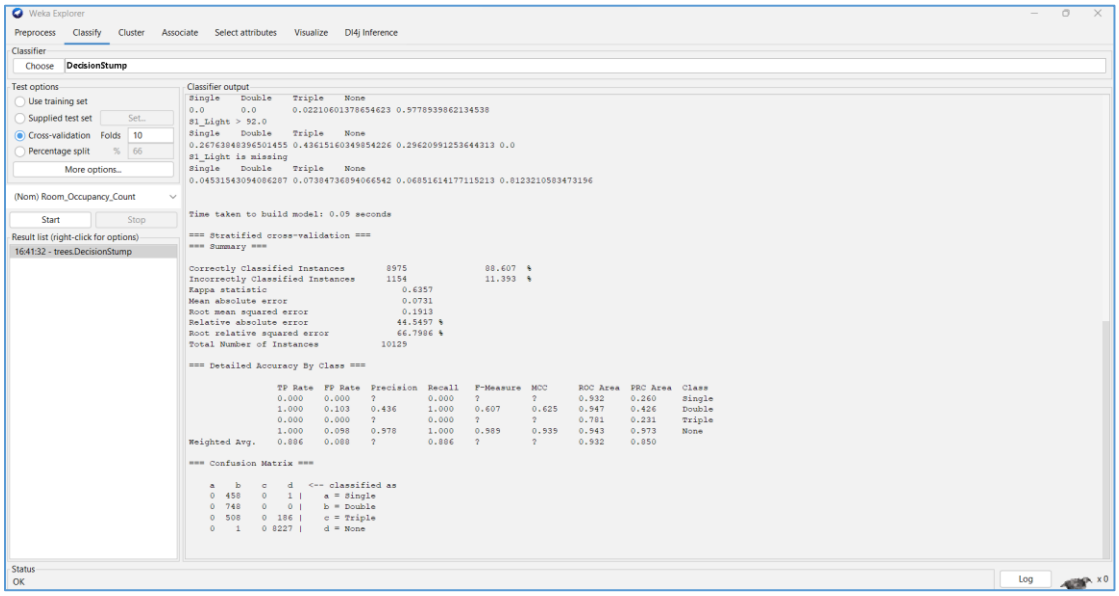


Figure 13: A screenshot of Weka (cross validation).

The Split Validation:

1- The Split Validation with 70% percentage Split:

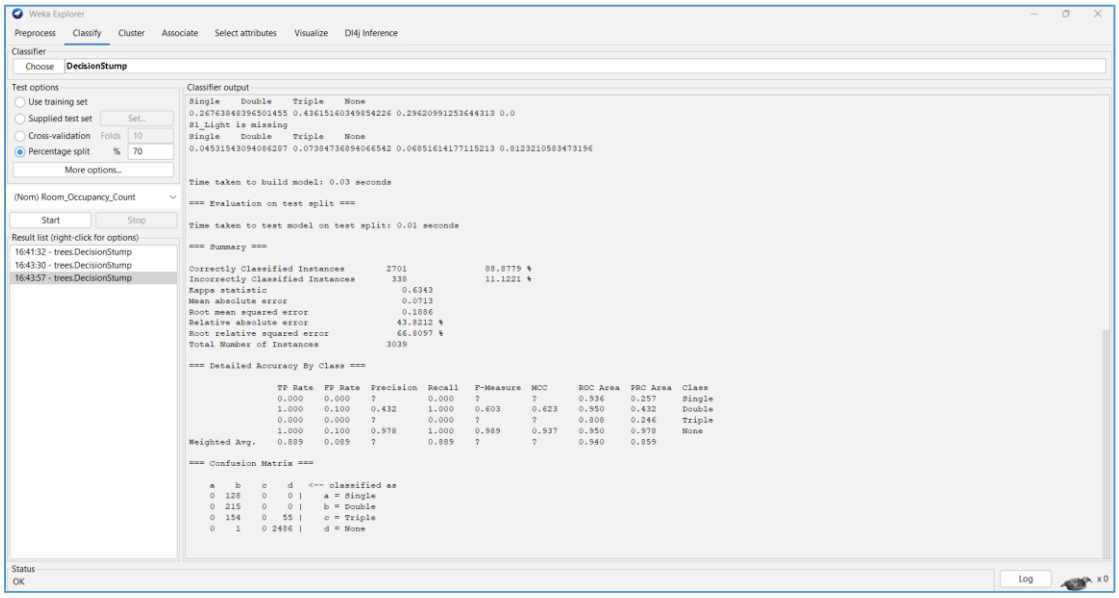


Figure 14: A screenshot of Weka (split validation with 70 percentage split).

2. The Split Validation with 80% percentage Split:

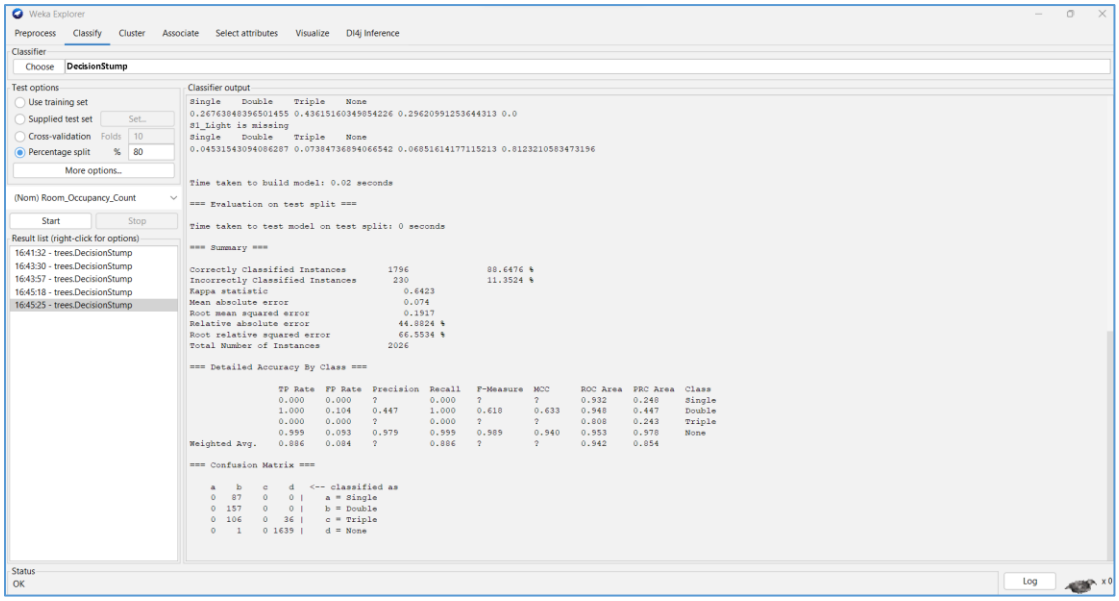


Figure 15: A screenshot of Weka (split validation with 80 percentage split).