University of Essex Online

MSc Artificial Intelligence

Final Submission:

Artificial Intelligence Solution Implementation

Submission date: 21.10.2024

Contents

Introduction

General Introduction

This assessment focuses on conducting experiments in WEKA, utilizing datasets from online resources. It is designed for the senior management of Investo, a fictional startup introduced in Unit 9 and aims to explore the feasibility of AI technologies in addressing the company’s operational needs. The report will not propose a precise solution using Investo's proprietary data but will instead leverage generalized datasets and models to contextualize AI's potential within the broader scope of the company's business objectives.

The assessment will be based on the CRISP-DM methodology, a widely recognized framework for structuring data mining projects. This methodology encompasses six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment (Wirth & Hipp, 2000). Given the nature of this assessment, the deployment phase will not be included, as the proposed solution will not be tested in a real-world application. The focus remains on the earlier phases to explore AI's potential within the business context.

Business Understanding

Investo is exploring AI-driven solutions to optimize various business processes, with customer segmentation being a primary challenge. The company faces difficulties in distinguishing between short-term customers, who engage with the product for testing purposes only, and long-term customers who invest larger sums over time, generating more revenue. They need a reliable solution to identify these customers early. AI can address this challenge by classifying customers into short-term and long-term groups. This segmentation will enable Investo to tailor marketing campaigns, gain better insights into long-term clients, and potentially convert short-term customers into loyal, long-term customers. They also have historical data labeled with customer types, which can be used to train a model.

AI Solution Approach

Data Understanding

The second phase of the CRISP-DM process, Data Understanding, involves gathering and thoroughly exploring the data. The primary objective is to become familiar with the dataset, identify quality issues, uncover hidden patterns, and detect relevant data subsets (Wirth & Hipp, 2000). For this assessment, a suitable dataset will be sourced from UCI or Kaggle, both of which offer diverse datasets. The goal is to select a dataset that clearly demonstrates AI's capacity for binary classification, distinguishing between long-term and short-term customers.

The selected dataset, available on Kaggle as the Mushroom Dataset (Binary Classification), boasts a usability score of 10.00, reflecting its high data quality. This score indicates that the dataset has passed Kaggle's checks for credibility, completeness, and compatibility, achieving 100% in each area. The dataset pertains to shop customers and includes 24’360 instances for class 0 and 29’675 instances for class 2.

Data Preparation

Hagendorff & Wezel (2020) highlight that obtaining the right data for model training is among the most significant challenges. Using biased, corrupt, or incorrect data can lead to highly unreliable AI systems. In the CRISP-DM methodology, the Data Preparation phase is critical for finalizing the dataset. This phase involves several key tasks, such as data cleaning, transformation, integration, and selection, ensuring the data is ready for modeling (Wirth & Hipp, 2000).

According to Nagashima & Kato (2019), data cleaning entails pre-processing steps like handling missing values, eliminating noise, removing duplicates, and correcting errors. Following this approach, the dataset has been reviewed for both completeness and correctness. To better align with the business case, the *Class* column has been modified from a numerical to a nominal value, representing customer types. All instances of 0 were relabeled as *short-term*, while all instances of 1 were relabeled as *long-term*.

The dataset did not require the integration of additional data sources, and all records will be used in their entirety. No further data selection or identification of subsets was necessary. For the modeling phase, the dataset will be split into training and validation sets following the 70/30 rule, a method that has been identified as highly effective for many machine learning algorithms by Nguyen et al. (2021). After obtaining the first model results, parameters will be adjusted to optimize accuracy. The data has been fully converted into the ARFF format to meet WEKA's requirements.

Modeling and Evaluation with WEKA

Justification of Algorithm Selection

The problem at hand is a classification issue, which can be tackled through various supervised and unsupervised machine learning algorithms. To offer Investo a well-rounded perspective on the machine learning landscape, both supervised and unsupervised methods will be employed for analysis.

Among the most widely recognized supervised classification algorithms is the Decision Tree (Charbuty & Abdulazeez, 2021), which will be applied to this problem. Another powerful supervised algorithm is Random Forest, which incorporates randomness into the traditional Decision Tree model. As noted by Biau & Scornet (2016), Random Forest is less prone to overfitting, making it a strong alternative to the Decision Tree for this classification task.

Presenting unsupervised learning algorithms to Investo's senior management could pave the way for future discussions on their value to the business. The K-Means clustering algorithm is one of the most used clustering algorithms in unsupervised learning. It will be applied to analyze Investo's historical data, segmenting it into k clusters, where k represents the number of customer segments. For this exercise, k will be set at 2, representing short-term and long-term customer segments (Khan, 2014). In practice, Investo could increase k to identify more natural groupings in the data. The K-Means algorithm has potential to uncover hidden patterns in Investo's customer base.

Decision Tree Algorithm Modeling and Evaluation

Among the decision tree algorithms available in WEKA is the J48 algorithm, a widely adopted method derived from the C4.5 algorithm (Bresfelean, 2007). For the purposes of this classification task, the J48 decision tree algorithm will be employed, using the labeled training dataset to facilitate accurate classification.

Upon importing the training dataset, containing 54,035 instances, into WEKA, the class attribute was designated as *Class*. Subsequently, the algorithm was configured to employ the J48 Decision Tree. Following the documented testing approach, the test options were set to a 70/30 percentage split, with 70% of the data used to train the model and 30% retained for testing and validation. The default parameters were used for model training. Under these conditions, the model successfully classified 98.174% of the instances in the test dataset, which consisted of 16,210 records, demonstrating high accuracy in classification.

Prior to conducting a more in-depth evaluation of the model, the testing parameters will be fine-tuned to optimize the accuracy and increase the number of correctly classified instances.

Parameter Change	Result
Percentage split 70%	98.174%
Percentage split 60%	97.9643%
Percentage split 50%	97.8828%
Percentage split 40%	97.5448%
Percentage split 80%	98.3622%
Percentage split 90%	98.1677%
Percentage split 85%	98.248%
Percentage split 75%	98.1716%
Percentage split 77%	98.3425%
Percentage split 76%	98.2264%
Percentage split 78%	98.326%
Cross-validation Folds 10	98.1679%
Cross-validation Folds 5	98.1142%
Cross-validation Folds 15	98.2141%
Cross-validation Folds 20	98.24%

In conclusion, the optimal results for the J48 Decision Tree algorithm were obtained using a 77% percentage split, resulting in 98.3425% of instances being correctly classified. The confusion matrix reveals that, among the 206 misclassified instances, 98 were incorrectly categorized as category B and 108 as category A, suggesting an even distribution of errors. Given the scope and limitations of this assessment, a deeper analysis of these misclassifications will not be pursued.

Random Forest Algorithm Modeling and Evaluation

As with the Decision Tree approach, the initial steps in constructing the Random Forest model involved uploading the dataset into WEKA and assigning the *Class* column as the target attribute during the preprocessing phase. Under the *Classify* tab, the *RandomForest* algorithm was selected, with all parameters set to their default values. Prior to model building, the test options were configured to a 70% percentage split. The first Random Forest model produced an accuracy of 99.0191%, correctly classifying 16,210 instances from the test set.

Before proceeding with the model evaluation, the testing parameters will be fine-tuned to optimize the accuracy and maximize the number of correctly classified instances.

Parameter Change	Result
Percentage split 70%	99.0191%
Percentage split 60%	98.9498%
Percentage split 50%	98.9451%
Percentage split 40%	98.8341%
Percentage split 80%	99.0839%
Percentage split 90%	99.0376%
Percentage split 85%	99.0993%
Percentage split 83%	99.0965%
Percentage split 87%	98.9609%
Percentage split 86%	98.9822%
Percentage split 84%	99.0978%
Cross-validation Folds 10	99.0025%
Cross-validation Folds 5	99.0043%
Cross-validation Folds 15	99.0562%
Cross-validation Folds 20	99.0321%

Following the adjustment of the testing options, while retaining the default algorithm parameters, the highest accuracy was achieved with an 85% percentage split, yielding a classification accuracy of 99.0993%. Analyzing the confusion matrix for further insights, it was observed that the model exhibited a tendency to misclassify instances into the B category, with 41 instances wrongly assigned to this category, compared to 32 instances incorrectly categorized as A.

K-Means Clustering Algorithm Modeling and Evaluation

The K-Means clustering algorithm, an unsupervised learning method, requires the removal of the class attribute from the dataset in the initial step. This algorithm operates by grouping the data into k clusters, where k is a predefined value (Sharma et al., 2012). For this assessment, k will be set to 2, aligning with the approach used in the previous supervised classification algorithms, to segment the data into two distinct categories.

Following the removal of the class attribute in the *Preprocess* step, the K-Means algorithm was selected from the *Cluster* tab in WEKA, with *SimpleKMeans* chosen as the specific algorithm. The Cluster mode was set to a 70% percentage split for the initial execution. This resulted in 5''447 instances (34%) being categorized into cluster 0, while 10'764 instances (66%) were grouped into cluster 1. Upon increasing the number of clusters to 4, the data distribution became more evenly spread, with 28%, 18%, 29%, and 25% of the instances allocated to the respective clusters.

To facilitate evaluation, the clusters will remain set at k = 2, while the percentage splits will be adjusted to explore different configurations.

Parameter Change	**Result**
Percentage split 70%	34% / 66%
Percentage split 60%	34% / 66%
Percentage split 50%	34% / 66%

Percentage split 40%	34% / 66%
Percentage split 90%	59% / 41%
Percentage split 99%	66% / 34%
Percentage split 30%	66% / 34%

The analysis showed that the 34% and 66% cluster split remained consistent across most dataset sizes tested, except for the 90% percentage split, where the data distribution between clusters was more even. These results imply that the dataset can be divided into two distinct groups, with approximately one-third of the instances assigned to one cluster and two-thirds to the other. Notably, the dataset had been labeled with a binary class that evenly divided the instances, suggesting that the unsupervised learning algorithm's different grouping may point to previously undiscovered patterns or hidden insights within the dataset.

Results

The findings indicate that both the Decision Tree and Random Forest algorithms can effectively classify Investo’s customers into long-term and short-term groups, given the high accuracy rates observed. For a successful implementation, Investo will need to provide a well-structured dataset containing customer information, account details, and labels indicating whether the customer is a long-term or short-term customer. The quality of the data will significantly influence the final model's accuracy, facilitating customer segmentation. Meanwhile, the K-Means clustering algorithm revealed that, even with a clean, evenly distributed binary class dataset, unsupervised learning can uncover hidden insights, suggesting alternative ways of clustering the data. Investo could leverage such algorithms to identify new customer segments that were not previously recognized.

Conclusion

The application of supervised and unsupervised learning algorithms has demonstrated that Investo's customer segmentation challenge can be addressed with AI. In particular, the supervised learning algorithms, Decision Tree (J48) and Random Forest, achieved very high accuracy in correctly classifying instances. Both algorithms can effectively segment customers into long-term and short-term groups, provided a high-quality dataset from Investo is available.

In contrast, the K-Means clustering algorithm, an unsupervised method, provided valuable new insights by forming clusters that suggest the discovery of hidden data patterns. Through unsupervised learning, Investo could identify previously overlooked customer groups and further refine their segmentation strategy.

In conclusion, the experiments illustrate the potential of AI in addressing complex problems while simultaneously generating new insights from data. Future steps for Investo could involve conducting further experiments with additional algorithms and datasets, as well as fine-tuning models. These efforts could provide Investo with a competitive advantage and maximize their return on investment.

References

Wirth, R. and Hipp, J., 2000, April. CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (Vol. 1, pp. 29-39).

Hagendorff, T., 2022. A virtue-based framework to support putting AI ethics into practice. Philosophy & Technology, 35(3), p.55.

Nagashima, H. and Kato, Y., 2019, March. APREP-DM: a Framework for Automating the Pre-Processing of a Sensor Data Analysis based on CRISP-DM. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (pp. 555-560). IEEE.

Nguyen, Q.H., Ly, H.B., Ho, L.S., Al-Ansari, N., Le, H.V., Tran, V.Q., Prakash, I. and Pham, B.T., 2021. Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, *2021*(1), p.4832864.

Charbuty, B. and Abdulazeez, A., 2021. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), pp.20-28.

Biau, G. and Scornet, E., 2016. A random forest guided tour. *Test*, *25*, pp.197-227.

Khan, Y.D., Khan, S.A., Ahmad, F. and Islam, S., 2014. Iris Recognition Using Image Moments and k-Means Algorithm. *The Scientific World Journal*, *2014*(1), p.723595.

Bresfelean, V.P., 2007, June. Analysis and predictions on students' behavior using decision trees in Weka environment. In *2007 29th International Conference on Information Technology Interfaces* (pp. 51-56). IEEE.

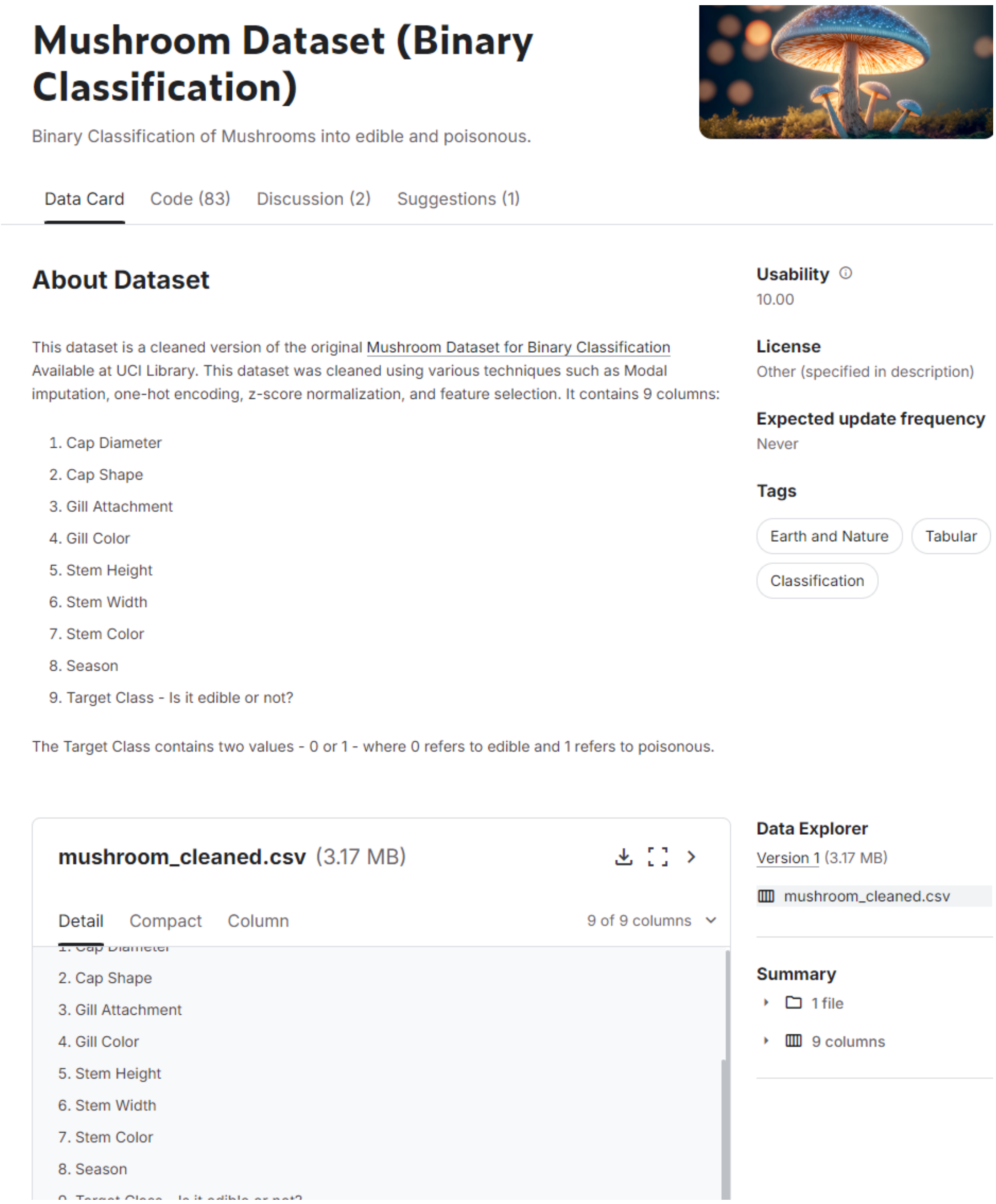
Sharma, R., Alam, M.A. and Rani, A., 2012, August. K-means clustering in spatial data mining using weka interface. In *International conference on advances in communication and computing technologies (ICACACT)* (Vol. 26, p. 30).

Appendix

This chapter will be used for screenshots of WEKA and all manual processed steps which are not already provided as part of the implementation documentation.

The dataset downloaded from Kaggle:

https://www.kaggle.com/datasets/prishasawhney/mushroom-dataset



Mushroom Dataset (Binary Classification)

Binary Classification of Mushrooms into edible and poisonous.

Data Card   Code (83)   Discussion (2)   Suggestions (1)

About Dataset

This dataset is a cleaned version of the original Mushroom Dataset for Binary Classification Available at UCI Library. This dataset was cleaned using various techniques such as Modal imputation, one-hot encoding, z-score normalization, and feature selection. It contains 9 columns:

1. Cap Diameter
2. Cap Shape
3. Gill Attachment
4. Gill Color
5. Stem Height
6. Stem Width
7. Stem Color
8. Season
9. Target Class - Is it edible or not?

The Target Class contains two values - 0 or 1 - where 0 refers to edible and 1 refers to poisonous.

Usability
10.00

License
Other (specified in description)

Expected update frequency
Never

Tags
Earth and Nature   Tabular   Classification

mushroom_cleaned.csv (3.17 MB)

Detail   Compact   Column   9 of 9 columns

2. Cap Shape
3. Gill Attachment
4. Gill Color
5. Stem Height
6. Stem Width
7. Stem Color
8. Season

Data Explorer
Version 1 (3.17 MB)
mushroom_cleaned.csv

Summary
1 file
9 columns

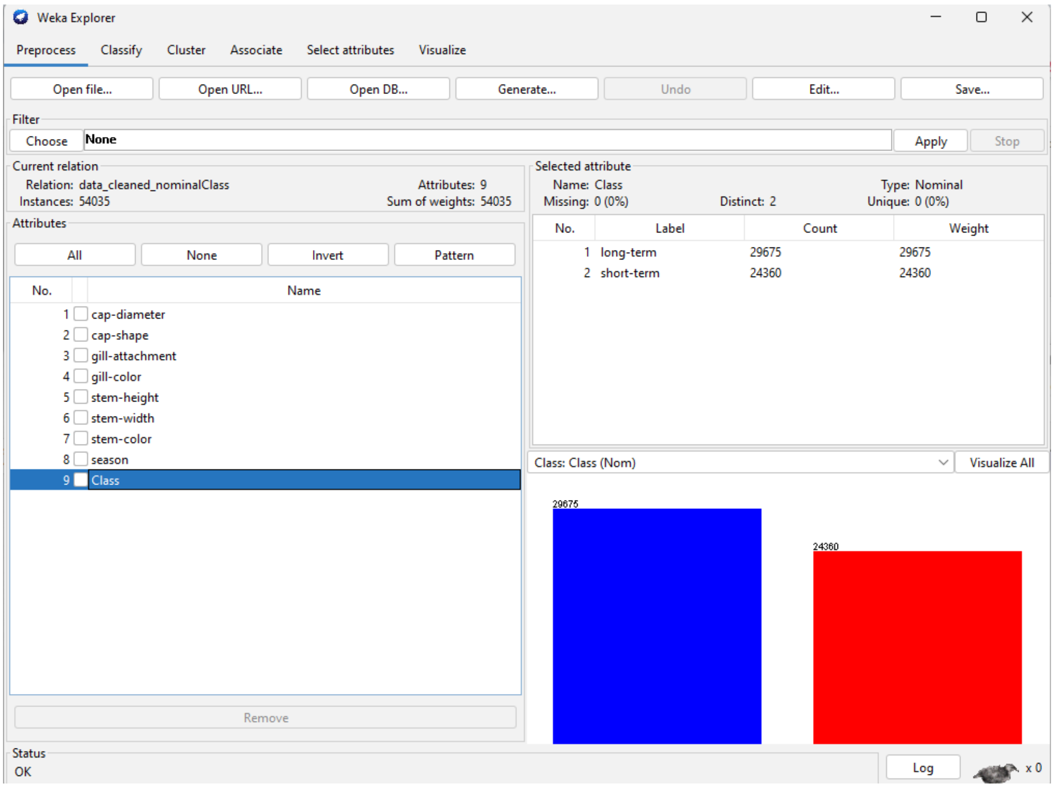
ARFF file has been created:

Name	Status	Date modified
CSV Files	✓	19.10.2024 16:16
mushroomCleanedNominalClass.arff	✓	19.10.2024 16:16

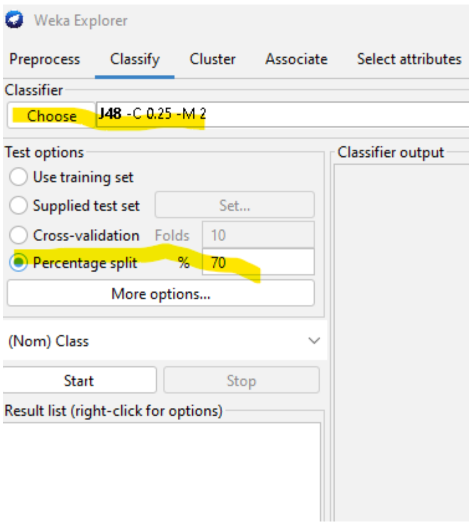
mushroomCleanedNominalClass.arff

```
@relation data_cleaned_nominalClass

@attribute cap-diameter numeric
@attribute cap-shape numeric
@attribute gill-attachment numeric
@attribute gill-color numeric
@attribute stem-height numeric
@attribute stem-width numeric
@attribute stem-color numeric
@attribute season numeric
@attribute Class {long-term,short-term}

@data
1372,2,2,10,3.807467,1545,11,1.804273,long-term
1461,2,2,10,3.807467,1557,11,1.804273,long-term
1371,2,2,10,3.612496,1566,11,1.804273,long-term
1261,6,2,10,3.787572,1566,11,1.804273,long-term
1305,6,2,10,3.711971,1464,11,0.943195,long-term
1337,6,2,10,3.775635,1520,11,0.943195,long-term
1300,2,2,10,3.83532,1563,11,1.804273,long-term
1354,6,2,10,3.67616,1532,11,0.88845,long-term
1222,6,2,10,3.771656,1476,11,0.943195,long-term
1085,6,2,10,3.775635,1581,11,0.88845,long-term
1214,6,2,10,3.696055,1524,11,1.804273,long-term
642,6,2,10,0.286062,1311,11,0.943195,long-term
814,4,2,10,1.189292,1681,11,0.943195,long-term
550,4,2,10,0.548675,1220,11,0.88845,long-term
```

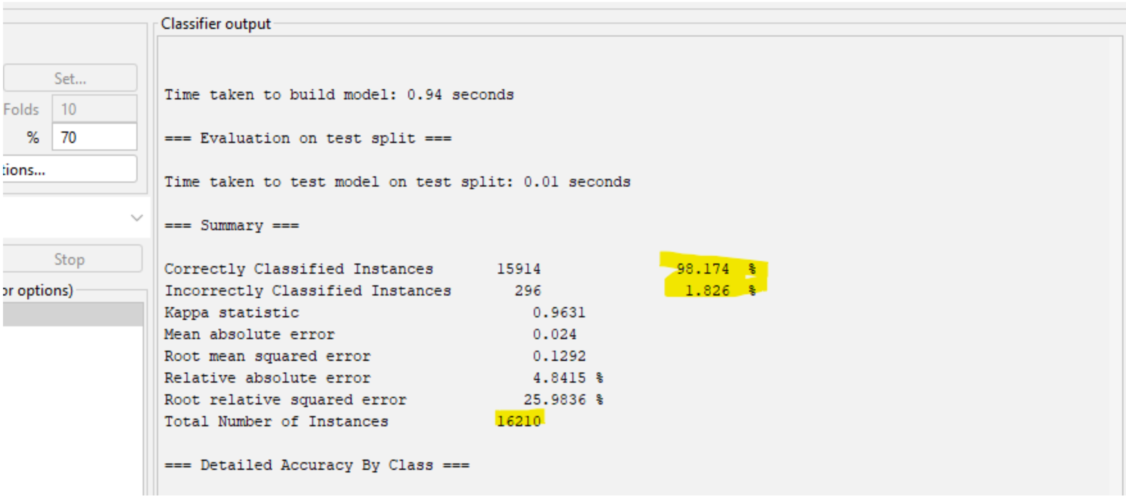
Ensuring the right class has been chosen:



Algorithm and Testing Selection:

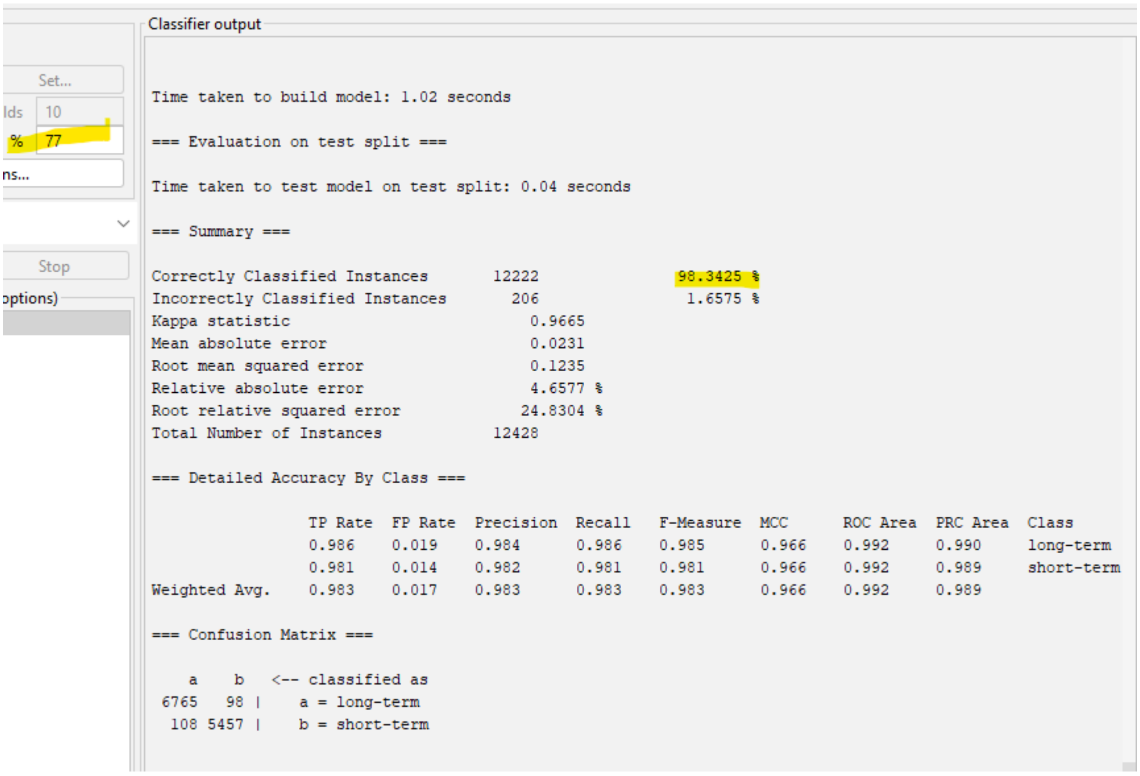


Decision Tree J48 run:




```
Time taken to build model: 0.94 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances       15914               98.174  %
Incorrectly Classified Instances       296                1.826  %
Kappa statistic                          0.9631
Mean absolute error                      0.024
Root mean squared error                  0.1292
Relative absolute error                  4.8415 %
Root relative squared error             25.9836 %
Total Number of Instances            16210

=== Detailed Accuracy By Class ===
```

Most accurate DT J48 run at a percentage split of 77%:




```
Time taken to build model: 1.02 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.04 seconds

=== Summary ===

Correctly Classified Instances       12222               98.3425 %
Incorrectly Classified Instances       206                1.6575 %
Kappa statistic                          0.9665
Mean absolute error                      0.0231
Root mean squared error                  0.1235
Relative absolute error                  4.6577 %
Root relative squared error             24.8304 %
Total Number of Instances            12428

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.019    0.984      0.986    0.985      0.966    0.992     0.990     long-term
                 0.981    0.014    0.982      0.981    0.981      0.966    0.992     0.989     short-term
Weighted Avg.    0.983    0.017    0.983      0.983    0.983      0.966    0.992     0.989

=== Confusion Matrix ===

    a    b   <-- classified as
 6765   98 |    a = long-term
  108 5457 |    b = short-term
```

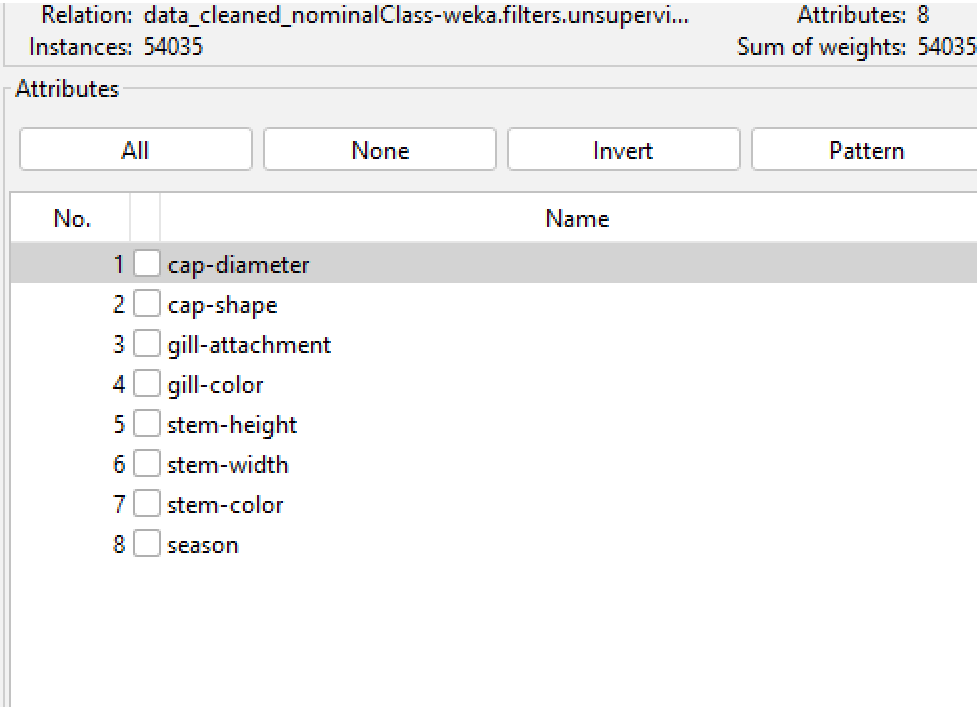
Random Forest first run with 70% percentage split test option and default parameters:

```
=== Summary ===

Correctly Classified Instances       16051               99.0191 %
Incorrectly Classified Instances       159                0.9809 %
Kappa statistic                          0.9802
Mean absolute error                      0.0289
Root mean squared error                  0.0959
Relative absolute error                  5.831  %
Root relative squared error             19.2868 %
Total Number of Instances            16210
```

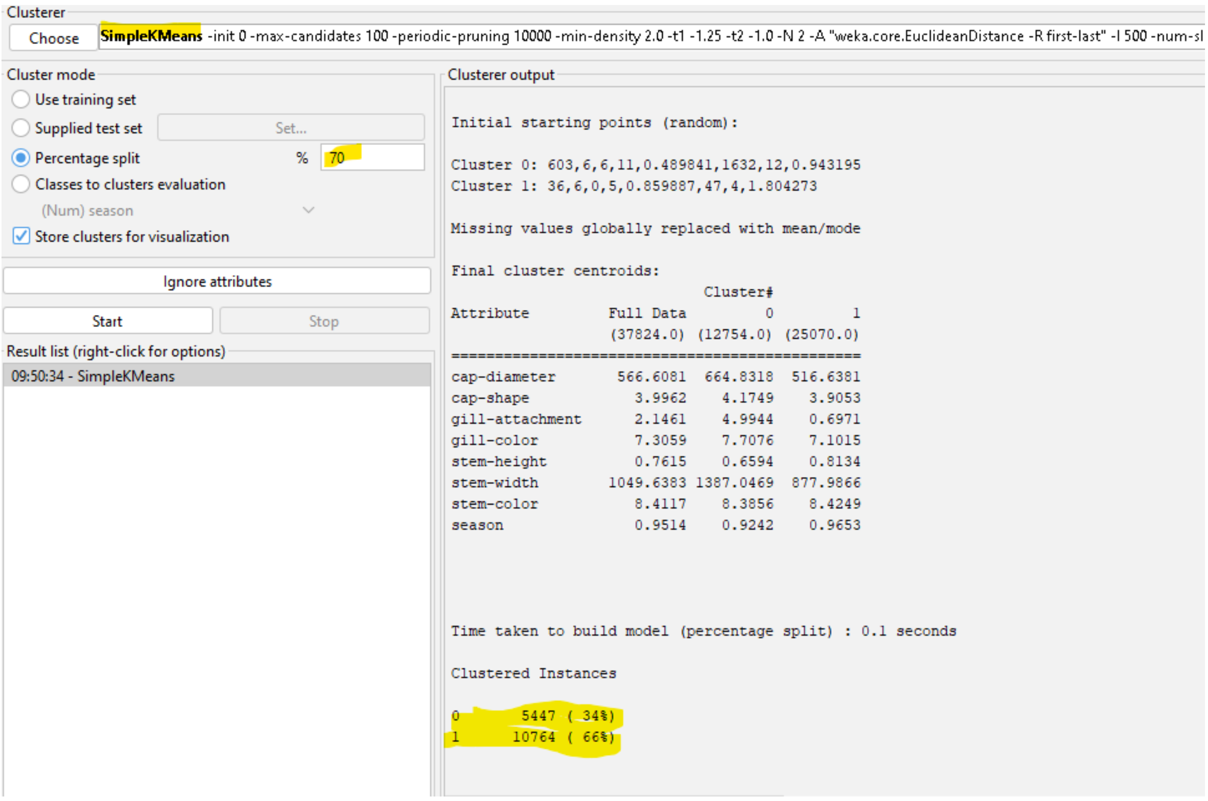
Best Random Forest model with a 85% percentage split test option and default parameters:

```
Classifier output
weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 10.62 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.25 seconds

=== Summary ===

Correctly Classified Instances        8032               99.0993 %
Incorrectly Classified Instances        73                0.9007 %
Kappa statistic                          0.9818
Mean absolute error                      0.0269
Root mean squared error                  0.0929
Relative absolute error                  5.4283 %
Root relative squared error             18.6918 %
Total Number of Instances             8105

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.991    0.009    0.993      0.991    0.992      0.982    0.999     1.000     long-term
                 0.991    0.009    0.989      0.991    0.990      0.982    0.999     0.999     short-term
Weighted Avg.    0.991    0.009    0.991      0.991    0.991      0.982    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 4439   41 |    a = long-term
   32 3593 |    b = short-term
```

Removed Class for unsupervised algorithm:



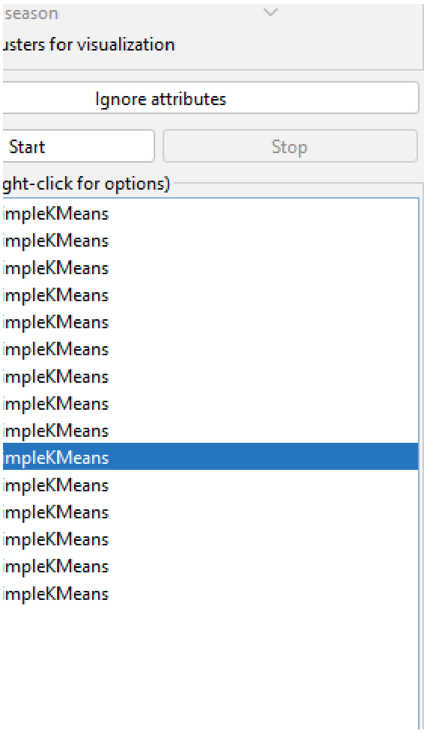
K-Means clustering run with percentage split on 70%



K-Means with 4 clusters

```
                              Cluster#
Attribute           Full Data         0          1          2          3
                    (37824.0)  (10529.0)   (7062.0)  (10524.0)   (9709.0)
==========================================================================
cap-diameter         566.6081   676.3964    451.838   671.8888   416.9087
cap-shape              3.9962     4.1711     5.1432     5.4803     1.3636
gill-attachment        2.1461     5.2339     1.8438     0.6879     0.5981
gill-color             7.3059     8.6631     3.9759     8.9165     6.5102
stem-height            0.7615     0.6768      0.777     0.7772      0.825
stem-width          1049.6383  1352.6044   896.7678   1186.641   683.7742
stem-color             8.4117     8.9203     5.4057     9.8187     8.5214
season                 0.9514     0.9489     0.9192     0.9852      0.941




Time taken to build model (percentage split) : 0.25 seconds

Clustered Instances

0       4523 ( 28%)
1       2908 ( 18%)
2       4649 ( 29%)
3       4131 ( 25%)
```

90% percentage split for K-Means clustering:




```
Missing values globally replaced with mean/mode

Final cluster centroids:
                              Cluster#
Attribute           Full Data         0          1
                    (48631.0)  (28278.0)  (20353.0)
===================================================
cap-diameter         566.9087   627.2264   483.1046
cap-shape              3.9976     5.7054     1.6249
gill-attachment        2.1428       2.13     2.1605
gill-color             7.3217     7.6624     6.8485
stem-height            0.7597      0.739     0.7884
stem-width          1051.3683  1186.5615   863.5338
stem-color             8.4141     8.5505     8.2245
season                 0.9522     0.9624      0.938



Time taken to build model (percentage split) : 0.12 seconds

Clustered Instances

0       3165 ( 59%)
1       2239 ( 41%)
```