**Data Mining and Machine Learning EBUS537**

**Assignment 1**

**by**

**Nina Daraio**

**201801820**

**10/12/2024**

**Contents**

* **Part 1**

Data Exploration **3**

Summary of Key Findings **3**

* **Part 2**

Tree Building **4**

Selection Process for Multiway Split **5**

* **Part 3**

Post Pruning **6**

Confusion Matrix **6**

* **Part 4**

Case Study **7**

* **References 10**
* **Appendix 11**

**Part 1**

**Data Exploration**

Data exploration of the car training dataset revealed important information about the structure of the data as well as the relationships between the variables present. The dataset contains 500 data entries across five variables, including categorical and numerical features. The class label “accept” appears imbalanced as most cars are classed as “not acceptable” (356) (*Appendix 1*), indicating a strict acceptability criterion. The data also contains some imbalanced variables such as “Doors” where most entries are heavily skewed towards 3-doors cars (*Appendix 2)*, and some balanced variable such as “persons” which is very balanced between 2 seaters (167), 4 seaters (169), and 5 seaters (164). *(Appendix 3)*

Relationships between variables show trends such as cars with a lower price and medium boot size seem to have a higher probability of being accepted, while cars with a high price and medium boot size are more likely to not be accepted. (*Appendix 4)* which might indicate that price and storage capacity are important decision metrics for choosing the right vehicle. Furthermore, this analysis reveals that cars designed for 4 or 5 “persons” have good acceptance rates as they can accommodate a larger number of people, compared to 2-seater cars (*Appendix 5)*.

Correlation analysis between “doors” and “persons” results in a correlation coefficient of -0.0365 indicating a very weak negative correlation, which suggests an opposite relationship: as the number of doors increases, the number of persons slightly decreases. Further analysis of “doors” shows a positive skewness (0.539) (*Appendix 6)*, while “persons” resulted in a negative skew (-0.376) (*Appendix 7)*, indicating a trend towards cars with a higher “persons” capacity possibly accommodating families or larger groups.

**Summary of Key Findings**

The above exploration demonstrates several key trends. The number of seats and boot size show a noticeable influence on the acceptability rate. The analysis also revealed distribution patterns and relationships among numerical attributes such as doors and persons, demonstrating the impact of car features on user preferences.

**Part 2**

**Tree Building**

To construct a fully grown decision tree to determine if a car is acceptable, Hunt’s Algorithm was implemented along with the Gini impurity measure using the Greedy Strategy.

The first step to create the tree is to evaluate the Gini impurity for each attribute and determine the root node, below is an example of how this was carried out.

**For Price:** Gini impurities need to be calculated for “price = high” and “price = low” (*Figure 1*). First it is necessary to count how many cars in the dataset are priced as “high”, storing the total in n1, the total for “low” is stored in n2.

p1 represents the total amount of times where “price = high” and “accept = yes” within the dataset. p2 represents the same for instances were “accept = no”. These values are then used within the Gini index formula to calculate the first Gini Child value.

The same calculation was carried out for cars where “price = low” and “accept = yes/no”. The two Gini Child values are then used together to calculate the Gini impurity for the “Price” attribute.

A screenshot of a computer code

Description automatically generatedThe same process was applied to all the other attributes and their respective categories, the results of the Gini impurity are in the table below (*Figure 2*). Then the attribute with the lowest Gini impurity is chosen as the root node, in this case “Persons” is the chosen attribute.

*Figure 1^*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Category** | **Gini Child** | **Weighted Gini** |
| **Price** | High | 0.3162006 | 0.3929551 |
| Low | 0.4722057 |
| **Doors** | 3 | 0.404824 | 0.4092312 |
| 4 | 0.3895098 |
| 5 | 0.4370447 |
| **Persons** | 2 | 0 | 0.3265095 |
| 4 | 0.4949407 |
| 5 | 0.4854253 |
| **Boot** | Medium | 0.3833775 | 0.4059788 |
| Big | 0.4597334 |

*Figure 2^*

**Selection Process for Multiway Splits**

The root node was identified to be “Persons”. The next step is to calculate Gini impurities for each category under “Persons”, filtering with the other attributes. For the category for “Persons = 2”, the Gini impurity was 0, resulting in a pure leaf node. This meant that no more splits are required for this subset and indicates that 2-seater cars are automatically not accepted.

Next, Gini impurities were calculated for the attributes “Price”, “Boot” and “Doors” where “Persons = 4”. Among these, “Price” had the lowest impurity measure (Gini = 0.5453315), compared to “Doors” (Gini = 0.492496) and “Boot” (Gini = 0.4923234). Following the greedy strategy, the tree was split on “Price”.

Further splits were made based on the values of “Price”. For “Price = low” and “Persons = 4”, Gini impurities were calculated for “Boot” (Gini = 0.4680397) and “Doors” (Gini = 0.4749121). “Boot” had the lowest impurity; therefore, this attribute guided the next split. For “Boot = Big” (Gini 0.4990548) Gini impurities were calculated for "Doors = 4" (Gini = 0.375), "Doors = 5" (Gini = 0.244898), and "Doors = 3" (Gini = 0.486111). For “Boot Med” (Gini = 0.4555248) the impurities were "Doors = 4" (Gini = 0.73), "Doors = 5" (Gini = 0.430), and "Doors = 3" (Gini = 0.459).

The same process was carried out for “Price = High”. Gini impurities were calculated for “Boot” (Gini = 0.3991369) and “Doors” (Gini = 0.4079954). Given the lower impurity measure, “Boot” was chosen as the next split. Based on this, impurity measures were calculated for “Doors = 3” (Gini = 0.345679), “Doors = 4” (Gini = 0.21875), and “Doors = 5” (Gini = 0.4844291) where “Boot = Med” as well as “Doors = 3” (Gini = 0.5), “Doors = 4” (Gini = 0.5), and “Doors = 5” (Gini = 0.4444444) where “Boot = Big”.

The same process was applied to the other side of the tree where “Persons = 5” which is then split by “Price”, “Boot”, and “Doors”, this process demonstrates the successful application of Hunt’s Algorithm.

The calculations were coded using R, and a sample calculation for one of the last nodes can be seen in Appendix 8.

**Part 3**

**Post Pruning**

With the fully-grown decision tree, post-pruning was applied to remove complexity and simplify the model. Pruning involves analysing all the subtrees where all the leaf nodes under their respective parent node result in the same class label. An example is below:

1. **“Persons = 4, Price = Low and Boot = Med”:**

The split for the attribute “Doors” resulted in:

* “Doors = 3: Yes (Gini = 0.4591837)”
* “Doors = 4: Yes (Gini = 0.4733728)”
* “Doors = 5: Yes (Gini = 0.4296875)”

After checking how many instances result in “no” and “yes” for each child node, a class label is assigned to each. In cases such as the one above where the class result is the same across all child nodes, the subtree was pruned by aggregating the class value. The parent node “Doors” was replaced by a single class node “Yes”. The full-grown decision tree can be found in Appendix 9

**Confusion Matrix**

A confusion matrix was created using the test dataset to compare predicted and actual class values, highlighting correct classifications and misclassifications (*Figure 3*). Key metrics such as accuracy, error rate, precision, recall, and F-score were then calculated to evaluate the model's performance (*Figure 4*).

|  |  |  |
| --- | --- | --- |
|  | **Predicted: Yes** | **Predicted: No** |
| **Actual: Yes** | 6 | 9 |
| **Actual: No** | 7 | 28 |

*Figure 3*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Accuracy** | **Error Rate** | **Precision** | **Recall** | **F - Score** |
| = (TP + TN)/Total ​= 6 + 28​/50 ​=**0.68** | = (FP + FN)/Total  = (7 + 9)/ 50  = **0.32** | = TP/(TP+FP)  = 6/(6+7)  = **0.4615** | = TP/(TP+FN)  = 6/(6+9)  = **0.4** | = 2\*Precision\*Recall/  Precision + Recall  = 2\*0.1846/0.8615  = **0.42855** |

*Figure 4*

The model achieved and accuracy of 68%, thus it was not able to correctly classify 32% of the test data. The low precision score suggests overprediction and the recall result shows that the model only identifies 40% of the “acceptable” cars which suggests a difficulty in detecting relevant cases. Incorporating additional attributes to the dataset might help the model classify cars more accurately.

**Part 4**

**Science Direct –** *Application of decision tree technology for image classification using remote sensing data.*

Decision tree-based classification models are often used in many different scenarios as they are easily interpretable, efficient and able to handle complicated data. The Cross Industry Standard Process for Data Mining or CRISP-DM model provides a structured approach for the implementation of complex data mining projects, ensuring consistency and clarity *(Saltz, 2021).* The following discussion analyses the case study titled *“Application of decision tree technology for image classification using remote sensing data”* which focuses on using hyperspectral data and decision tree algorithms to classify images for agricultural purposes. This analysis will be conducted through the CRISP-DM framework with the aim to demonstrate how decision trees can simplify classification processes.

The first stage in the CRISP-DM model is about the business understanding, this entails understanding the objectives of the project and any limitations *(Schröer et al., 2021).* Within the case study the aim was identified to be the use of decision tree for the classification of agricultural plots based on tillage practices, residue levels and cropping practices to help precision farming. Decision trees were selected as they can efficiently process large amounts of input data and provide clear decision rules which can be critical for interested stakeholders. The study also outlines important challenges such as ensuring the models’ robustness to any data variations and addressing the need for cost-effective management solutions, aligning with the goals of the business understanding phase.

Understanding the data being used is key when selecting an effective classification method. This study uses hyperspectral data, which was collected across 71 wavebands, capturing reflectance from visible to near-infrared spectra for 18 agricultural plots. This hyperspectral data is especially useful within the agriculture sector as it is capable of capturing reflectance patterns that change based on factors such as crop conditions. For the correct implementation of a decision tree classification method, the data utilised must be correctly pre-processed. To reduce noise the hyperspectral data underwent a series of corrections such as geometric and atmospheric. The two formats of input data were: reflectance values for all 71 wavebands and normalised difference vegetation indices. This step was necessary to help the decision tree model to process data efficiently and achieve high classification accuracy.

Next, the CRISP-DM model goes through the modelling phase which entails applying classification methods to build predictive models. In this case classification and regression trees (C&RT) were used. C&RT models work by dividing data into subsets based on decision rules deriving from the input variables. Decision trees are suitable for this context as they can extract human readable rules, critical for implementation in precision agriculture. For this case study, models were developed to help the classification of different tillage practices, residue levels and cropping practices. Parameter tuning including adjustments to tree depth and minimum node size, ensured that the models avoided overfitting. Cross-validations was also used to make sure that the evaluation was robust. The aim of the decision tree model was to extract understandable classification rules as well as identifying wavebands which differentiated between classes.

Next the evaluation phase measures the effectiveness of the classification methods employed. Within this case study decision trees achieved high classification accuracies, particularly for residue levels and tillage practices. Logistic regression models were also used for comparison. Misclassification metrics showed that decision trees were effective in distinguishing between agricultural classes. These evaluation methods aligned with the business objectives set in the first stage *(Schäfer et al., 2018)*.

The deployment phase of CRISP-DM was not explicitly addressed within the case study although it could entail the integration of the decision tree models into precision farming. For example, classification rules could guide framers in selecting appropriate management practices.

For this case study following the CRISP-DM model proved useful as it ensured an organised approach to all stages of the CRISP-DM model.

**References**

* Saltz, Jeffrey S. “CRISP-DM for Data Science: Strengths, Weaknesses and Potential next Steps.” *2021 IEEE International Conference on Big Data (Big Data)*, 2021.
* Schäfer, Franziska, et al. “Synthesizing CRISP-DM and Quality Management: A Data Mining Approach for Production Processes.” *IEEE Xplore*, 1 Nov. 2018.
* Schröer, Christoph, et al. “A Systematic Literature Review on Applying CRISP-DM Process Model.” *Procedia Computer Science*, vol. 181, no. 1, 2021, pp. 526–534, www.sciencedirect.com/science/article/pii/S1877050921002416, https://doi.org/10.1016/j.procs.2021.01.199.
* Yang, Chun-Chieh, et al. “Application of Decision Tree Technology for Image Classification Using Remote Sensing Data.” *Agricultural Systems*, vol. 76, no. 3, 1 June 2003, pp. 1101–1117, www.sciencedirect.com/science/article/pii/S0308521X02000513#FIGGR2, https://doi.org/10.1016/S0308-521X(02)00051-3.

**Appendix**

A graph with a green rectangle

Description automatically generated *Appendix 1*

A graph with orange squares

Description automatically generated*Appendix 2*

A graph of a number of red rectangular objects

Description automatically generated

*Appendix 3*

A graph of a number of different colored bars

Description automatically generated with medium confidence*Appendix 4*

A graph of a graph of a car capacity

Description automatically generated with medium confidence*Appendix 5*

A graph with a line and a red line

Description automatically generated*Appendix 6*

A graph with a green line

Description automatically generated*Appendix 7*

*A screenshot of a computer code

Description automatically generated*

*A white background with text and numbers

Description automatically generated with medium confidenceAppendix 8*

*A diagram of a flowchart

Description automatically generatedAppendix 9*