Car Evaluation Dataset

Gagandeep Batra
Applied Machine Learning in Data Science.

Abstract

Cars take a part in our everyday lives. There are many different types of cars and a buyer has a difficult choice to make. Each different aspect in a car makes up different factors to consider. The dataset utilized in this case study takes those factors and predicts the acceptability value of the car. This case study uses Decision Tree and Naive Bayes models to compare the performance between them. The results illustrate the Naive Bayes model performing better than Decision Tree in all aspects.

1 Introduction

There are many different factors to consider when buying a car. All cars are different due to type, make, model, and manufacturer. Standard equipment is one of the main factors to consider when buying a car. It defines the most important features in daily life, such as safety. This case study takes into account convenience features as well as safety to solve important impactful problems such as accident number reduction.

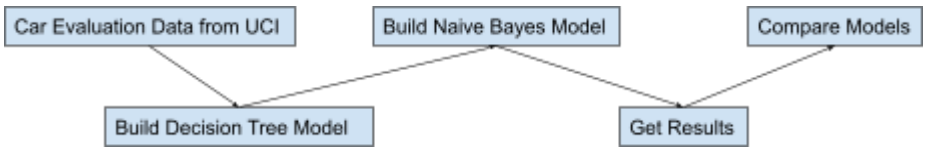
There are many similar works that utilize the Car Evaluation dataset. For example, in [2], they build Artificial Neural Network (ANN) and Naive Bayes models to compare. From their tests, they found Naive Bayes to perform consistently across each one. They were effective in comparing the different algorithms and ineffective in observing more tests.

Classification is a branch of Machine Learning which uses supervised learning to categorize a set of data into classes. It is most commonly used today in speech recognition, face detection, and many more. The application of classification uses algorithms such as Decision Tree, Naive Bayes Classifier, and many others.

1.1 Our Contribution

- Classification Accuracy of Naive Bayes Classifier
- Classification Accuracy of Decision Tree
- Comparison of Naive Bayes Classifier and Decision Tree

2 Material and Methods



There were 5 methods used in this case study. After getting the data, the main method was to build the Decision Tree and Naive Bayes models. Lastly, the results from the models are compared to find the most efficient algorithm.

2.1 Dataset Description

The dataset used in this study is multivariate which includes a collection of 1728 instances on different attributes of cars. It was donated by Marko Bohanec in 1997 and was obtained from the UCI Machine Learning Repository [1]. The Car Evaluation Dataset was derived from a simple hierarchical decision model and is depicted in Table 1.

Table 1: Car Evaluation Dataset

Data Set Characteristics:	Multivariate	Number of Instances:	1728
Attribute Characteristics:	Categorical	Number of Attributes:	6
Associated Tasks:	Classification	Missing Values?	No

The class values in the Car Evaluation Dataset are:

- Unacceptable: denoted with ‘unacc’
- Acceptable: denoted with ‘acc’
- Good: denoted with ‘good’
- Very Good: denoted with ‘vgood’

The attributes in the Car Evaluation Dataset are:

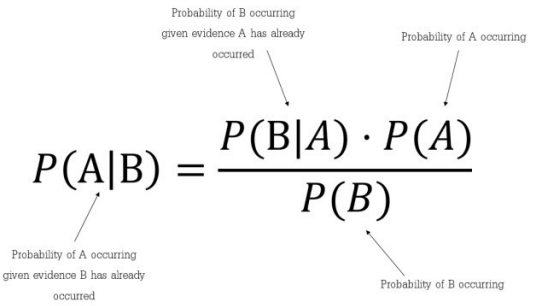
- Buying (‘buying’): vhigh, high, med, low
- Maintenance (‘maint’): vhigh, high, med, low
- Doors (‘doors’): 2, 3, 4, 5more
- Persons (‘persons’): 2, 4, more
- Luggage Boot (‘lug_boot’): small, med, big
- Safety (‘safety’): low, med, high

The dataset uses these attributes to classify a car as unacceptable, acceptable, good, or very good. These features were chosen to provide overall car acceptability based on price, technical characteristics, comfort, and safety.

2.2 Algorithms Used or Methodology

The Naive Bayes Classifier is classification technique based on Bayes' Theorem (seen in Figure 1) with an assumption of independence among predictors [3]. It captures uncertainty using a probabilistic approach. By using the Bayes' Theorem, the probability of a class value can be calculated given the attributes. The largest probability between the class values will be chosen as the predicted target. The parameters will be based on the class values and attributes from the dataset. Naive Bayes was used for this case study because encoding the probabilities can be useful for finding the occurrences for each individual event.

Figure 1: Bayes' Theorem



https://www.kdnuggets.com/wp-content/uploads/bayes-nagesh-1.jpg

Decision Tree is a non-parametric supervised learning method used for classification tasks. It is a tree-like graph with each instance sorted down from the root to some leaf node which represents the class label for the instance. The final result involves decision nodes and leaf nodes. The model predicts the best attribute based on information gain, a measure of how well an attribute splits that data into groups based on classification [4]. Decision Tree was used for this case study because it is a non-parametric method which can be used to compare with a parametric method such as Naive Bayes.

2.3 Performance Evaluation

In a dataset, a training set is implemented to build up a model, while the testing set is used to validate the model built. For this case study, the data set used for training is from the same set as testing. The three splits used in this study can be seen in Table 2. The learning method is based on the attributes and class values of the training set. From that model, values are predicted for the testing set to find the efficiency for the algorithm.

Table 2: Car Evaluation Data Split for Model

Training	Testing
90%	10%
60%	40%
50%	50%

3 Experimental Analysis

The supervised model is built based on the class values in correspondence to the values. The accuracy achieved from Decision Tree and Naive Bayes can be seen in Tables 3 and 4, respectfully. Each table includes the accuracy and inaccuracy in relation to the training and testing split.

Table 3: Classification Accuracy of Decision Tree

Training / Testing Split		Decision Tree	
Training %	Testing %	Correct %	Incorrect %
90%	10%	79%	21%
60%	40%	79%	21%
50%	50%	78%	22%

Table 4: Classification Accuracy of Naive Bayes

Training / Testing Split		Naive Bayes	
Training %	Testing %	Correct %	Incorrect %
90%	10%	86%	14%
60%	40%	82%	18%
50%	50%	81%	19%

Naive Bayes outperformed Decision Tree in each of the different splits. The 90% training and 10% testing split provided the best accuracy for both of the classification methods. On the other hand, the 50% training and 50% testing provided the worst accuracy. Overall, from the results, Naive Bayes seems to be the better model for the dataset. The code for each test can be seen on GitHub [https://github.com/gsBatra/CarEvaluationData].

4 Conclusion

This case study analyzed the performance of Decision Tree Classifier and Naive Bayes on the Car Evaluation dataset from UCI. In terms of accuracy, the results found Naive Bayes performing better than Decision Tree in all the tests. The limitations of this case study include not enough data to fully train and only 2 models were utilized. Possible future work may include analyzing more models to find the best one for this dataset.

References

[1] UCI Machine Learning Repository: Car Evaluation Data Set. (1997). UCI. https://archive.ics.uci.edu/ml/datasets/Car+Evaluation.

[2] Performance Comparison of Data Mining Algorithms: A Case Study on Car Evaluation Dataset. (2014). IJCTT. https://www.researchgate.net/profile/Jamilu-Awwalu/publication/287397675_Performance_Comparison_of_Data_Mining_Algorithms_A_Case_Study_on_Car_Evaluation_Dataset/links/595a4d01458515a5406fc4b8/Performance-Comparison-of-Data-Mining-Algorithms-A-Case-Study-on-Car-Evaluation-Dataset.pdf

[3] 1.9. Naive Bayes — scikit-learn 0.24.2 documentation. (n.d.). Scikit-Learn. https://scikit-learn.org/stable/modules/naive_bayes.html.

[4] 1.10. Decision Trees — scikit-learn 0.24.2 documentation. (n.d.). Scikit-Learn. https://scikit-learn.org/stable/modules/tree.html.