Car Evaluation Analysis

Deepthi Suryadevara
TSYS College of Computer Science
Columbus State University
Columbus, GA, USA
suryadevara_deepthi@columbusstate.edu

Prudhvi Raj Nelapatla
TSYS College of Computer Science
Columbus State University
Columbus, GA, USA
nelapatla_prudhviraj@columbusstate.edu

***Abstract*—Car is an integral part of our lives. While we spend measurable time in driving, most of us cannot survive without it. Evaluating the car is hence an important task before buying as it ensures the safety, security, and convenience it provides. As the manual classification is time consuming and labor intensive, a machine learning model can automate the process of classification based on the cars attributes and help us choose the best one in much less time. One of the main applications of Machine learning is classification. This inspired the authors choose building a good machine learning model which considers all the features of the car and decide whether it is acceptable or not. A decision tree classifier and a Neural Network (Multi-Layer Perceptron) are built and optimized further to achieve best possible results. The database we choose has four classes(class) and hence we must build a multi-class classification model. We first performed exploratory data analysis (EDA) and preprocessed the data. We then trained models and tested using test data and checked for accuracy. After training the models using the best chosen hyperparameters, the multi-layer perceptron model achieved highest accuracy.**

Keywords—decision tree classifier, artificial neural network, machine learning models, multi-layer perceptron

I. INTRODUCTION

In today's digital world, we are constantly dealing with a large amount of information which makes the study of data, a crucial task and extremely necessary. As the human brain is incapable of studying such vast amounts of information, we need machines to process such big data. Buying cars is a very cumbersome task as it involves considering various factors like safety, luxury, and cost. These factors vary for different cars based on manufacturer, model, and type of the car. Convenience, safety features and performance play an important role in car selection. Choosing a best car with good features reduce the accidents. For this research we chose to work car evaluation dataset and build a classifier on it to correctly predict the car acceptability. Previously many algorithms were used by other researchers to classify this dataset like Random Forest, Linear Perceptron, KNN, Nayes Bayes algorithm, Decision Tree, etc. which gave an accuracy of around 85 to 95. Our interest is to build a Decision Tree Classifier and an Artificial Neural Network and compare the results for accuracy and aim to get ideal efficiency of 100%. As we wished to use decision tree classifier and neural network, we studied few papers where they used these algorithms to solve similar kind of problems.

The paper, 'Prediction of MS Graduate Admissions using Decision Tree Algorithm' helps assessing the Test attributes like GRE, TOEFL, CGPA, Research papers published etc. and predicts the eligibility of Indian students getting admission in best university in US. They achieved this using decision tree algorithm for predicting the output in terms of possibility of admission in different universities based on their scores and helps the students choose the best university. The dataset they used has 500 rows and 9 features. While performing EDA, they found that the major issues were missing and inconsistent data. They used pairplot and heatmap to visualize the data. By training using decision tree and predicting results, they achieved 93% accuracy. The results were good at predicting the enrolment with a good accuracy using very few features related to the student. The limitations as per the authors is the authenticity of the data set and inability to capture SOP, LOR, etc. which need advanced modelling (NLP). The accuracy of the model they developed will be reduced they considered SOA and LOR. Hence there is a scope of further improvement of their research.

The paper, 'Spam Detection Using Artificial Neural Networks (Perceptron Learning Rule)' deals with the bulk emails sent with no agreement between the sender and the receiver to receive email. A spam filter is needed to prevent the delivery of these spam emails. This is a serious problem since the legitimate emails are cluttered by spam and the existence of email services is threatened. In this paper, the researchers discussed a technique to identify spam artificial networks and perceptron learning rule. Spam causes innumerable problems to all the people. They discussed how it cost you time, invades privacy, damages the "end to end" principle, and creates problems to Internet Service Providers. They talked about some of the research about the current state of the art spam prevention techniques, legislation to create an enforceable law to impose heavy penalties and the filtering techniques. Usually, the spam filtering occurs at the end of the SMTP transaction which is content filtering. Some of the popular content filters are Rule Based Filters, Bayesian Filters, Support Vector Machines, and Artificial

Neural Networks. One more type of spam filtering is network level filtering which blocks a blacklist consisting of spam IP addresses. They then discuss the perceptron algorithm, training algorithm and learning algorithm adapted from Alia et al. [4]. They performed experimentation, training and testing on their perceptron and employed a stochastic gradient method for training purpose. "For each training value, the perceptron continuously uses said value until it either generates the desired output or reaches a pre-specified maximum number of iterations."[2] Error and weights are calculated after each iteration by comparing the output with the needed output and update the weights. This process is repeated until the stop condition is met. The error is calculated by sum of squares error equation. The desired learning rate and maximum iteration number are determined by experimenting in the test environment and then the testing was done on the messages which were not used for training. The perceptron algorithm yielded a very good spam detection rates as it incorporates continuous learning feature. Hence, this cannot be easily overcome by spammers. This algorithm can be further improved to block server identification.

II. Dataset Description

Car Evaluation Data Set was derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.)[6]. The data set contains examples with the structural information removed, i.e., directly relates the target concept, CAR to the six input attributes: buying, maint, doors, persons, lug_boot, safety instead of the three intermediate concepts: PRICE, TECH, COMFORT. It contains 1728 car sample data with six attributes describing the characteristics of a car and one class feature that tells about the car condition.

The following are the details of all the attributes:

- class: Unacceptable(unacc), Acceptable(acc), Good(good), VeryGood(vgood)
- buying: Buying Level or Capacity of the customer (Very High: vhigh, High: high, Medium: med, Low: low)
- maint: Maintenence Level (Very High: vhigh, High: high, Medium: med, Low: low)
- doors: Number of doors in the car (2, 3, 4, 5 or more)
- persons: Capacity in terms of to carry (2, 4, and more) persons
- lug_boot: The size of the Luggage Boot (small, med, big)
- safety: Safety Level of Car (low, med, high)

Since the dataset is multi class, we built a multilayer feed forward model with back propagation algorithm and a multi class decision tree classifier.

III. Methodology

A decision to buy a car or not based on its characteristics must be made. The car evaluation dataset has six attributes based on which each car is to be classified into unacceptable, acceptable, good or very good.

A Decision tree model is a supervised learning method which learns from the data labels. A predictive model is generated from several binary rules in tree form to classify the desired variable. Since it can visualize, it becomes easy to understand decision hierarchy and relations between features. Since it needs very little preprocessing of data, it is easy to build this model and faster to get results. But even minor changes in data cause complete change of tree structure. It also as overfitting problem and less likely to get desired accuracy compared to a neural network. The basic algorithm of a decision tree is that we first create a root node, then calculate the entropy of each attribute and select the attribute that has the highest information gain which classifies best. Next, is to bind that branch to the root node and repeat this process until the entropy is zero or information gain is either zero or one.

ANN is a network of artificial neurons which send signals by means of connection links like the real neurons in the brain. Links have weights which are multiplied by network input. An activation function is applied to the net input to calculate the output. In a Feedforward Neural Network, inputs are directly connected to the outputs with a single layer of weights. In this network, all the neurons are directed towards the front. Each neuron on the layer is connected to another neuron on the next layer without feedback connection [4]. Multilayer Feedforward Neural Network consists of multiple inputs at the input layer, hidden layers (one or more), and the outputs at the output layer where error is calculated. This Desired-Actual Output error is used to modify the weights to get minimum error. The input data is sent in the forward direction from one layer to another up to the output layer and error from output layer is propagated backward to tune the hidden layers which minimizes the gradient. This network is called Multi-Layer Perceptron model.

We first perform exploratory data analysis (EDA) and preprocessed the data. We then built a decision tree classifier model and a neural network model (multi-layer perceptron classifier) and trained both models with the training data. We predicted the class of the test data using the built models. The results of both models are checked for accuracy and compared. We then optimized both the models by GridSearchCV estimator which will search for the best set

of hyperparameters. We also plot the decision tree. Bolow are the detailed steps involved in the model development of the two types of algorithms, i.e., decision tree and ANN.

A. *Load and Clean Data*

The raw data is first imported into a pandas data frame and cleaned to ensure the quality. We renamed the columns for better understanding of data and then checked for missing data and the values from each column. By plotting all the columns, we understood that all are categorical. Hence, we created the categorical data types to convert the categorical values.

	vhigh	vhigh.1	2	2.1	small	low	unacc
0	vhigh	vhigh	2	2	small	med	unacc
1	vhigh	vhigh	2	2	small	high	unacc
2	vhigh	vhigh	2	2	med	low	unacc
3	vhigh	vhigh	2	2	med	med	unacc
4	vhigh	vhigh	2	2	med	high	unacc

Fig. 1. Original Sample Raw Data

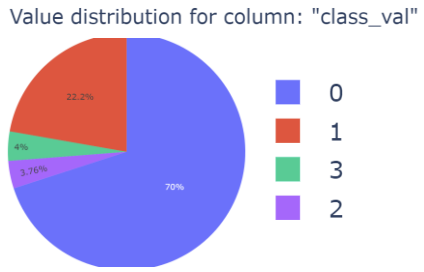


Fig. 2. Class Variable Distribution

B. *Data Preprocessing*

This is an important task before using data as it prepares and makes the data ready for better analysis, visualization, and modelling. We converted categories into integers for each column, 0 representing least preferred or valued and increase denoting more valued.

	buying	maint	doors	persons	lug_boot	safety	class_val
0	3	3	0	0	0	1	0
1	3	3	0	0	0	2	0
2	3	3	0	0	1	0	0
3	3	3	0	0	1	1	0
4	3	3	0	0	1	2	0

Fig. 3. Final Sample Dataset (Cleaned and Preprocessed)

C. *Dataset Split*

The six feature attributes and the target variable (class_val) are selected. We then split the preprocessed data available into train and test sets in the ratio of 70:30 using train_test_split. Only one split is used in this research.

D. *Train and Test*

A part of the dataset i.e., the training dataset is used to train the algorithms and the test data is used to test the built models to test for accuracy. For the decision tree, we first initialized a decision tree estimator with maximum depth of 5 and criterion as entropy and then trained the estimator. We then plot the tree seen in Fig. 4. which is a simplified version.

```
|--- safety <= 0.50
|   |--- class: 0
|--- safety >  0.50
|   |--- persons <= 0.50
|   |   |--- class: 0
|   |--- persons >  0.50
|   |   |--- buying <= 1.50
|   |   |   |--- maint <= 1.50
|   |   |   |   |--- safety <= 1.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- safety >  1.50
|   |   |   |   |   |--- class: 3
|   |   |   |--- maint >  1.50
|   |   |   |   |--- safety <= 1.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- safety >  1.50
|   |   |   |   |   |--- class: 1
|   |   |--- buying >  1.50
|   |   |   |--- maint <= 1.50
|   |   |   |   |--- lug_boot <= 0.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- lug_boot >  0.50
|   |   |   |   |   |--- class: 1
|   |   |   |--- maint >  1.50
|   |   |   |   |--- buying <= 2.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- buying >  2.50
|   |   |   |   |   |--- class: 0
```

Fig. 4. Decision Tree

We then predicted the output variable of the test data. A 10-fold cross validation is used which performs the fitting procedure ten times. Each time, 90% of the total training set is selected randomly while the leftover 10% can be used for validation. This is a good preventive measure to overcome overfitting problem which is most common with decision trees.

For MLP we first initialized a multi-layer perceptron classifier with 5 hidden layers and 1000 iterations and trained it. We then tested this model with the test data and compared with the actual data for checking accuracy of the built model. We also performed a 10-fold cross

validation as for decision tree model and calculated accuracies of both models.

E. Tune Hyperparameters and Repeat Train and Test

We used the GridSearchCV module to search for the best set of hyperparameters which improve both models. For decision tree classifier, the hyper parameters to be checked are criterion and maximum depth. Of the two criterions gini and entropy, gine is the best and with maximum depth of 11 chosen best out of 1 to 20. The following MLP hyper parameters are checked:

‘activation’: 'logistic','tanh','relu'
‘solver’: 'lbfgs','adam','sgd'
'alpha':10.0 ** -np.arange(1,3)
'hidden_layer_sizes': (5),(100),(3),(4),(3,1),(5,3)

The best one’s chosen by GridSearchCV are: activation function is tanh, solver is lbfgs, alpha is 0.1 and hidden layer size is 100. The models are trained again with the new set of hyper parameters and tested for accuracy. Accuracy, f1 score, ROC, and Confusion matrix are calculated for each model for comparison of efficiencies.

IV. Results

After testing the models with the test data, the predictions are compared to the actual values to measure the efficiency of the models. To assess the models, we used metrics such as mean squared error (MSE), mean average error (MEA), cross validation accuracy and accuracy. We also plot the confusion matrix as it is best to understand the classification issues as we can know the true positives, true negatives, false positives, and false negatives and understand where the model went wrong. We can achieve this by calculating positive predicted value and negative predicted value. The comparison of efficiencies of the decision tree, optimized decision tree, neural network and optimized neural network can be viewed in the Fig. 6. We can see that the best hyperparameters yielded the best models and the neural network performed best compared to decision tree. The cross validation improved the models and searching for best hyperparameters further improved. If time is also an issue to consider, then we can say decision tree performed reasonably in less time compared to neural network.

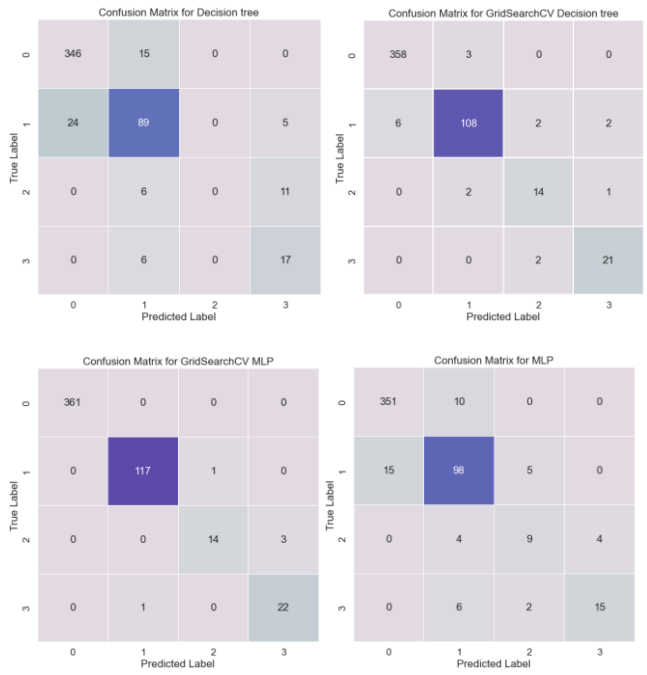
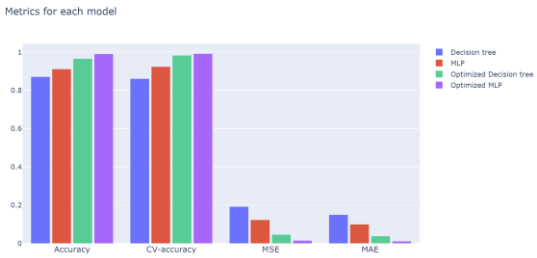


Fig. 5. Confusion Matrices



	Accuracy	CV Accuracy	MSE	MAE
Decision Tree	0.87	0.86	0.19	0.15
Optimized Decision Tree	0.97	0.98	0.05	0.04
Neural Network (MLP)	0.91	0.92	0.12	0.10
Optimized Neural Network (MLP)	0.99	0.99	0.02	0.01

Fig. 6. Results Plot and Table of Accuracies

From the accuracies table in Fig.6. we see that Optimised MLP achieved 99% accuracy which is almost ideal value. Hence this is the best model.

V. Conclusions

This study allowed for a better understanding of how these algorithms provide a faster and less complex way to classify the data. A decision tree classifier and a Neural Network (Multi-Layer Perceptron) are built and optimized further to achieve best possible results. This research is a comparative analysis of both models which shows that Multilayer Perceptron of Artificial Neural Network (ANN) takes longer to build and test model compared to Decision Tree and the 10-Folds Cross Validation. However, in terms of accuracy, the Multilayer Perceptron with tuned hyperparameters with 99% accuracy is the best. Further research can be done to analyze confusion matrices to reduce all the false values. Further research can be done to analyze the confusion matrices to completely get rid of the false predicted values.

VI. Acknowledgments

We would like to thank our professor, Dr. Rania Hodhod for her timeless support, encouragement, and

advisement in the completion of this project. We also thank and give credit to UCI data repository and Marco Bohanec and Kaggle for providing this dataset available to use.

REFERENCES

[1] Janani P, Hema Priya V, and Monisha Priya S, "Prediction of MS Graduate Admissions using Decision Tree Algorithm", Volume 9 Issue 3, March 2020. www.ijsr.net, Licensed Under Creative Commons Attribution CC BY. DOI: 10.21275/SR20307225734. https://www.ijsr.net/archive/v9i3/SR20307225734.pdf

[2] Owen Kufandirimbwa, Richard Gotora, "Spam Detection Using Artificial Neural Networks (Perceptron Learning Rule)". ISSN 2315-5027; Volume 1, Issue 2, pp. 22-29; June 2012. Online Journal of Physical and Environmental Science Research. https://tarjomefa.com/wp-content/uploads/2016/08/5143-English.pdf

[3] Pravarti Jain, and Santosh Kr Vishwakarma, "A Case Study on Car Evaluation and Prediction: Comparative Analysis using Data Mining Models", International Journal of Computer Applications 172(9):21-25, August 2017. https://www.ijcaonline.org/archives/volume172/number9/28279-2017915205

[4] Sucheta Chauhan, Prof. Prema K. V, "Car Classification Using Artificial Neural Network", International Journal of Scientific and Research Publications, Volume 2, Issue 12, December 2012. ISSN 2250-3153. http://www.ijsrp.org/research-paper-1212/ijsrp-p1240.pdf

[5] Jamilu Awwalu, Anahita Ghazvini, and Azuraliza Abu Bakar, "Performance Comparison of Data Mining Algorithms: A Case Study on Car Evaluation Dataset", International Journal of Computer Trends and Technology (IJCTT) – volume 13 number 2 – Jul 2014. https://www.ijcttjournal.org/Volume13/number-2/IJCTT-V13P117.pdf

[6] Yegnanarayana, "Feedforward Neural Networks" in Artificial Neural Networks, New Delhi, India: Prentice- Hall of India, 1999, ch. 4, pp. 88-141. https://www.ijcaonline.org/archives/volume172/number9/jain-2017-ijca-915205.pdf

[7] https://towardsdatascience.com/car-evaluation-analysis-using-decision-tree-classifier-61a8ff12bf6f

[8] https://towardsdatascience.com/detecting-a-simple-neural-network-architecture-using-nlp-for-email-classification-f8e9e98742a7

[9] https://www.kaggle.com/datasets/elikplim/car-evaluation-data-set