Assignment 4
A mathematical essay on Decision Tree

Anik Bhowmick
Inter Disciplinary Dual Degree Data-Science
Indian Institute of Technology Madras
ae20b102@smail.iitm.ac.in

***Abstract*—This assignment discusses the application of the Decision Tree Classifier on the Car Evaluation Data Set. This data 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.). The data comprises various kinds of car features, such as the buying price, maintenance price, maximum passenger capacity, safety, etc. Based on it, the task is to predict whether a car is acceptable or not, depending on its safety. This data is completely categorical, so its preprocessing involves various feature handling techniques such as label encoding, one hot encoding, Missing value imputations, application of the Chi-squared test for deriving the correlations, etc.**

***Index Terms*—Visualization, Decision tree classifier, Label encoding, Cramer's V rule, Correlation coefficient, Confusion Matrix, Accuracy, Precision, Recall, F$_1$ score, ROC.**

I. Introduction

- This assignment is a multiclass classifier about the acceptance of a car based on its safety parameters using the Decision Tree classifier. The dataset used for this purpose is the Car Evaluation Data Set. The main task is to unfold the underlying features in the data that decide the rank of the acceptance of a car.
- A Decision tree is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict whether an instance belongs to a given class or not. They work by recursively splitting the dataset into subsets based on the most significant feature, creating a tree-like structure of decisions. There are several ways to decide which feature is the most important to split: Gini Index, Entropy, etc. Each internal node in the tree represents a decision based on a specific feature, and each leaf node represents the predicted outcome or class label. The working principle is consistent with its name.
- The main task in this assignment is to build a multiclass classifier model to classify acceptable cars from unacceptable ones by looking into their various attributes; for this purpose, the decision tree algorithm will be used. A decision tree works very well when the attributes are categorical in nature.
- This paper will demonstrate the data analysis technique relevant to the decision tree classifier with the help of visual plots and mathematical equations. This paper covers various data handling techniques, the model's fitting and validation, the algorithm's working principle, and the mathematics behind various evaluation metrics.

II. Decision Tree

As the name suggests, a Decision tree works by splitting the data into a tree-like structure. The bottom-most parts are called the leaves, which contain the class labels. Some terminologies related to the decision tree are given below:

- **Root Node:** The topmost node in the tree represents the entire dataset as the root node.
- **Splitting:** Dividing a node into sub-nodes based on a certain attribute or feature.
- **Decision Node:** A node in the tree that has children nodes, indicating a decision is made based on the feature's value, is known as a decision node.
- **Leaf (Terminal) Node:** A node in the tree with no children represents the final decision or the output class.
- **Branch/Sub-Tree:** A section of the decision tree, including a parent node and its descendants.
- **Attribute/Feature:** The property used to make decisions at each node.
- **Splitting Criterion:** The measure used to split nodes, such as Gini index, Entropy, or Information gain.
- **Pruning:** The process of removing branches that do not provide significant information, improving the tree's accuracy on unseen data.
- **Overfitting:** Creating a decision tree that is too complex and performs well on the training data but poorly on new data.
- **Underfitting:** Creating a decision tree that is too simple to capture the underlying patterns in the data.

A. Assumptions

- Non-linearity: Decision trees assume a non-linear relationship between features and the target variable. They can model complex, non-linear patterns in the data.
- Feature Relevance: Decision trees assume that the data-splitting features are relevant to the prediction task. Irrelevant or redundant features can affect the tree structure and may lead to overfitting.

- Local Optima: Trees make locally optimal decisions, not always globally optimal. Because it adapts a top-down greedy approach.
- Recursive Binary Splitting: Data is recursively split into subsets at each node to two different categories.
- Splitting Criteria: Information Gain, Gini Index, and Entropy are used by decision trees to split the data at each node.

Decision trees work with both categorical and numerical features through appropriate preprocessing methods for categorical data and the inherent ability to handle numerical comparisons. This flexibility contributes to their widespread use in various machine-learning applications.

B. Attribute Selection Measures

Training of a decision tree involves splitting the original dataset into subsets based on Attribute Selection Measures. The ASM is done to ensure as large homogeneity as possible in the resulting subset data; this maximizes (Information Gain) or minimizes (Entropy/ Gini index) the attribute selection criterion. This process is repeated recursively until there is no further improvement in the criterion. This kind of training does not involve any trainable model weights/ parameters or domain knowledge. This feature enables decision trees to work for very large dimensional data.

1) Entropy: It is a measure of the impurity in a particular feature. Let us denote $p_{k|c}$ as the probability of data points of a unique category of a given attribute belonging to the kth class. The entropy is defined as follows:

$$p_{k|c} = \frac{\text{Total data points of class k in that category}}{\text{Total number of data points in that category}}$$

$$H(D|\text{Attribute} = \text{nth category}) = -\sum_{k=1}^{k=K} p_{k|c} log_2(p_{k|c})$$

To calculate the entropy of the entire attribute, we take a weighted average of individual entropies across all the unique categories. Suppose there are n different categories in that attribute:

$$p_n = \frac{\text{Total number of data points in nth category}}{\text{Total number of datapoints in the attribute}}$$

$$H(D|\text{Attribute}) = \sum_{n=1}^{n=N} p_n H(D|\text{Attribute} = \text{nth category})$$

The information gain measures how much entropy has dropped due to splitting under an attribute.

$$IG = \text{Total entropy of dataset} - \text{Entropy of a feature}$$

The total entropy of the dataset can be calculated as:

$$p_k = \frac{\text{Total number of datapoints in class k}}{\text{Total number of data points in the dataset}}$$

$$H(D) = -\sum_{k=1}^{k=K} p_k log_2(p_k)$$

There is one more measure of impurity called the Gini Index. A Gini index of 0 indicates perfect purity, where all the elements belong to a single class, while a Gini index of 1 signifies maximum impurity, where the classes are evenly distributed. The Gini index can be given as:

$$\text{Gini} = 1 - \sum_{k=1}^{k=K} (p_k)^2$$

C. Evaluation Metrics

Evaluation metrics are useful for the assessment of a model after its training.

- **Confusion Matrix** A confusion matrix is the table often used to describe the performance of a classification model on a set of test data for which the true values are known. A confusion matrix looks exactly as given below.

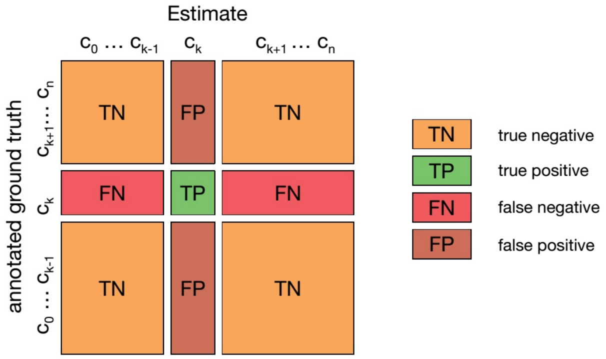

Fig. 1. Confusion matrix for multiclass classification

  TP: True positive is how many positives are true as predicted by the model.
  FP: False positive is how many are predicted to be positive but not positive.
  FN: False negative is how many are falsely predicted to be negative.
  TN: True negative is how many are predicted to be negative, which are actually negative.
  It is clear that only TP and TN are the correct predictions made by the model. The rest are wrong predictions. A good model should have FP and FN as small as possible.
- **Accuracy** We define accuracy as the fraction of correct prediction out of the total prediction given by the model.

  $$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

  A good model should have high accuracy. But that is not a necessary condition. Even a highly accurate model can give more wrong predictions if trained on a highly imbalanced dataset (i.e., the number of one class instances is much larger than the other). So, accuracy alone can not help us decide whether a model is good or not.

- **Precision**: This is defined as how many are actually positive out of total positive prediction.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall**: or true positive rate is defined as the number of predicted positive out of actual positive. It is also known as Sensitivity.

$$\text{Recall (TPR)} = \frac{TP}{TP + FN}$$

  For multiclass classification for each class, precision and recall can be defined. Increasing recall decreases precision, often known as ”precision-recall trade-off”. It depends from problem to problem what we want: more precision or more recall. Like in the medical field of cancer disease detection models, we want the model to become robust to detect cancer. In this case, predicting a negative, even if a patient is positive, is highly undesirable. So, we want a more positive rate, meaning high recall. But in the present data, we need not look into these details. Instead, we can check the respective F_1 scores.
- ***F*₁ score** : Is the harmonic mean of precision and recall.

$$\boldsymbol{F_1} \ \textbf{score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Again, for F_1 score for each class, it can be defined.

III. The Data

Brief information about the data:

- Total data points are 1728. Columns are ’Buying price’, ’Maintenance cost’, ’Number of doors’, ’Number of persons’, ’lug_boot’, ’safety’, and ’decision’. Please note the original data didn’t have any column names. So, we had to add these names on our own based on the information available.

Variable	**Definition**	**Key**
buying	buying price	vhigh, high, med, low
maint	Price of the maintenance	vhigh, high, med, low
doors	Number of doors	2, 3, 4, 5, more
persons	Capacity in terms of persons to	2, 4, more
lug_boot	The size of luggage boot	small, med, big
safety	Estimated safety of the car	low, med, high
Target	Target variable to predict	unacc, acc, good, vgood

TABLE I
Description of the data

IV. The Problem

As mentioned, our goal in this assignment is to predict whether a car is safe, acceptable, or unacceptable.

A. Cleaning and preparing the data

1) Encoding the categorical columns: Because any ML model can’t work with strings, we need numerical representations of the categorical features. There are two kinds of encoding primarily used in Machine learning tasks. One hot encoding and Label encoding. In our case, we decided to continue with the label encoding technique. One advantage is that it does not unnecessarily increase the number of features corresponding to each unique instance. For a small dataset like this, an increased number of features might lead to overfitting of the model.

B. Exploratory analysis

1) Correlation among the numerical featurers: Two features are correlated if they have an absolute correlation coefficient close to 1. They are uncorrelated if their correlation coefficient is closer to 0. For categorical features, the conventional person correlation technique can not be applied. There is a technique called Cramer’s V correlation is widely used for this purpose. It ranges from 0 to 1, indicating no association to a perfect association, respectively. First, the contingency table is calculated. It lists frequencies or counts of the combinations of every two categorical variables. Then the chi-squared test is performed.

$$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Here O_{ij} is the observed frequency in each cell, and E_{ij} is the expected frequency in each cell. The Cramer’s V rule is given as :

$$V = \sqrt{\frac{\chi^2}{n \times \min(k - 1, r - 1)}}$$

Where n is the total number of observations, k is the number of categories in a variable, and r is the number of categories in another variable.

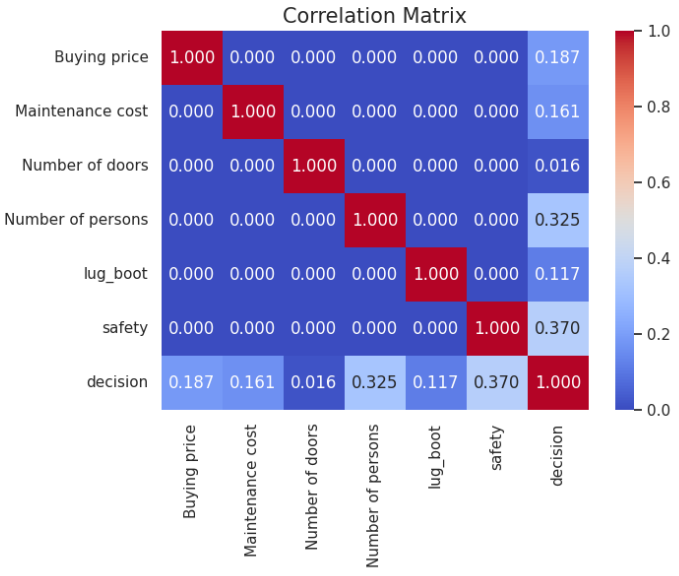

Fig. 2. Correlation plot

All the input categorical features are uncorrelated because the values of correlation coefficients are zero. So, there is no need to drop any column.

2) Preliminary study on the trend followed by data: We will study a few plots before finally feeding the data to the model.

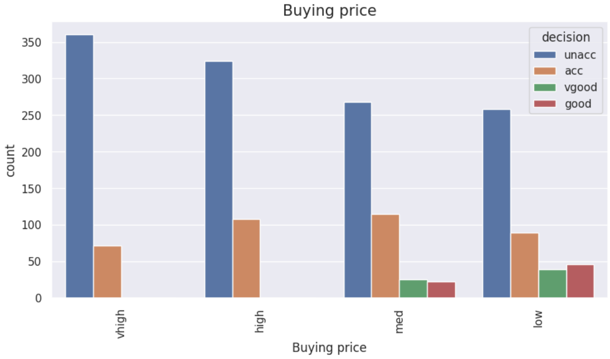

Fig. 3. Buying price and acceptability

From this plot, it is clear that expensive cars have lesser acceptability. Whereas cheaper cars are more under acceptable, good, and very good class. The number of unacceptable cars is much less than that of expensive cars.

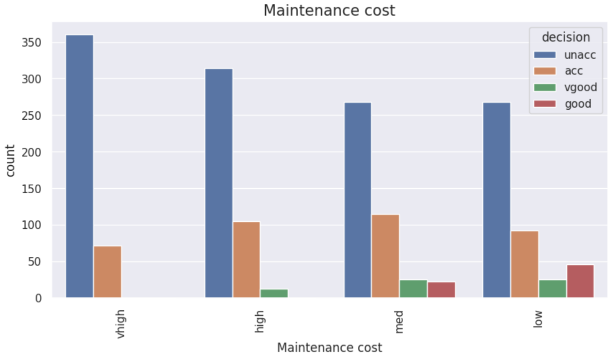

Fig. 4. Acceptabilty with maintenance cost

Cars with low maintenance costs are rejected comparatively less than those that take huge maintenance costs. It is evident from this plot.

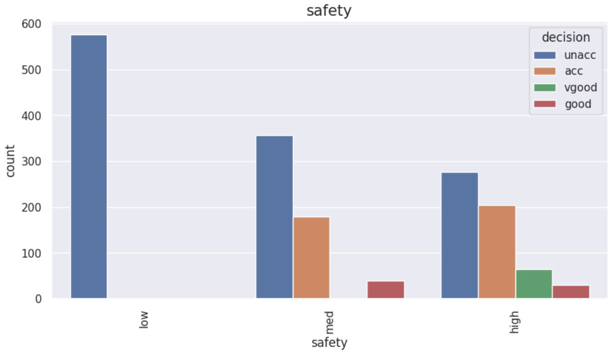

Fig. 5. Acceptance with car safety

Common intuition suggests that, safer the car, the more it will be accepted. And unsafe cars will be less acceptable. The same is reflected in the above plot. Here, unsafe cars have no acceptability at all.

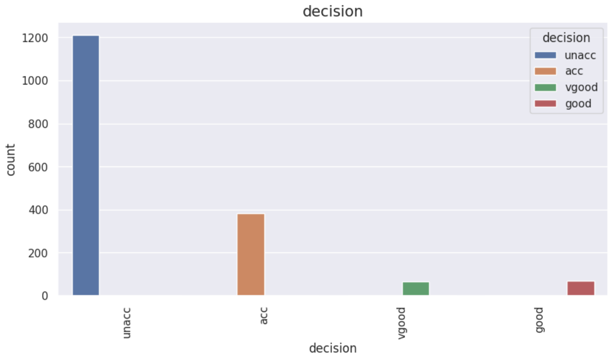

Fig. 6. Absolute count of 4 classes

This plot clearly conveys the fact that classes are highly imbalanced, with the unacceptable class being the largest in number. This gives a clear indication that accuracy can not be used for the model validation. Other metrics, such as precision, recall and F_1 score, need to be used. The true counts of these 4 classes are unacc: 1210, acc: 384, good: 69, vgood: 65.

C. *Splitting the Training data to Train and validation sets*

To validate the model, the data will split into training and test sets with test size being 20%. Because of the imbalanced dataset, it is highly necessary to make stratified splits. Because it is never desirable that the training set does not contain any minority classes viz, good and vgood at all, and all the good, vgood classes are only present in the test set. Naturally, the model outputs will be full of errors, and the model might not even see the minority classes during training. So, stratification is very important for this dataset. This ensures both the train and test set contain all the classes with almost similar ratios.

D. *Statistical model*

Our model is a Decision tree classifier from sklearn. The criterion used for training was Gini; after testing with various depths, the best depth was found to be 12. The model scored fairly well on both the train and test sets. The train set accuracy was 99.8%, and the test set accuracy was 99.1%. This clearly indicates even with depth 12; the model did not encounter over-fitting. The detailed evaluation metrics are given in the next section

E. *Visualization and validation*

Below are the evaluation metrics for the validation set:

Classes	Precision	Recall	F_1 Score
Acc	0.97	0.99	0.98
Good	0.93	1.00	0.97
Unacc	1.00	1.00	1.00
Vgood	1.00	0.92	0.96

TABLE II
EVALUATION METRICS

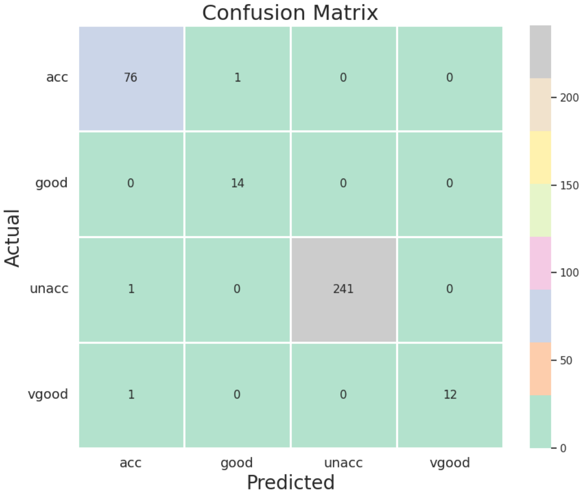

Fig. 7. Confusion Matrix with heatmap

In this confusion matrix for multiclass case, all the diagonal numbers are the correct predictions made by the model. Non-diagonal terms are all incorrect predictions. The model with a lesser value of non-diagonal entries is the best one. The false positive, true positive, false negative and true negative notions are somewhat vague in the context of multiclass classification but still can be defined. From this figure, we can infer the following

- For acc class, true and predicted labels are 76, and the number of true other class labels that are predicted to be of acc class is 2. So precision is 76/78=0.97. Similarly, recall is 76/77=0.99.
- For class good, true and predicted labels are 14, and the number of true other class labels that are predicted to be of good class is only 1. So precision is 14/15=0.93. There are no other predicted classes with ground truth being good. So, a recall of 1 is achieved.
- For class unacc, true and predicted labels are 241; no other true class labels are predicted to be of unacc class. So precision is 1. There is only 1 other predicted class other than unacc, which belongs to unacc actually. So, a recall of 0.995 is achieved for this class.
- Similarly, for the vgood class, achieved precision and recalls are 1 and 0.92, respectively.

From the confusion matrix, we get the accuracy as:

$$\text{Accuracy} = \frac{\text{Sum of diagonal terms}}{\text{Sum of all elements}}$$

$$\text{Accuracy} = \frac{76 + 14 + 241 + 12}{76 + 14 + 241 + 12 + 1 + 1 + 1} = 0.9913$$

The goodness of a model can be tested by another metric called area under the ROC (Receiver operating characteristics) curve. The more the area, the better the model is. In the present scenario case, AU-ROC values for each of the classes are 0.99 for class Acc, 1.00 for class Good, 1.00 For class Unacc and 0.92 for class Vgood. Please note that Acc denotes Acceptable, Unacc implies Unacceptable, and Vgood is Very good. So, with all this analysis, the model performance is undoubtedly extremely good.

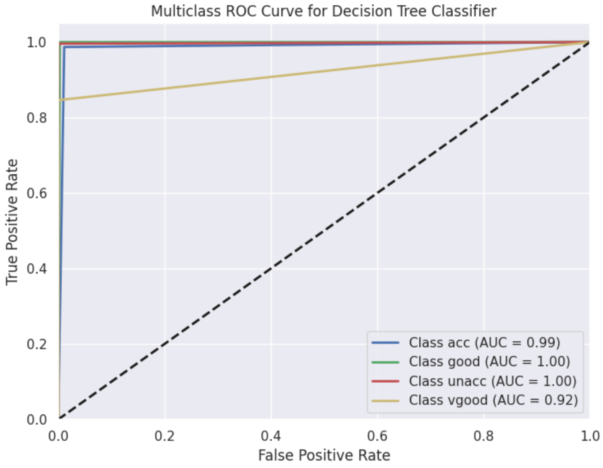

Fig. 8. Receiver operating characteristic curve

1) Insights in test data and visual validation of model: Bar plots are analysed in the test set to gain insights from the data. In addition to that, this visualization will help assess the model performance manually.

- The following plot is the acceptance of a car based on the purchase price. This plot is similar to the one of the exploratory data analysis. The number of unacceptable cars is the highest in all the price ranges. The acceptance of cars is the least in the case of the most expensive ones. This is quite obvious. This trend is illustrated in the same way in both the original labelled data and the predicted data.

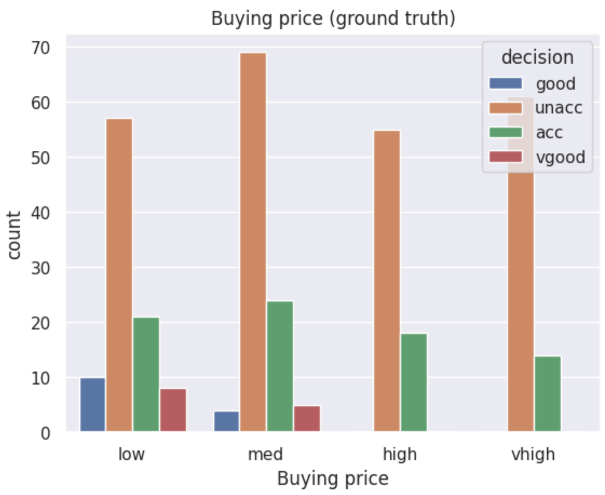

Fig. 9. Acceptabilty of the car based on buying price (original data)

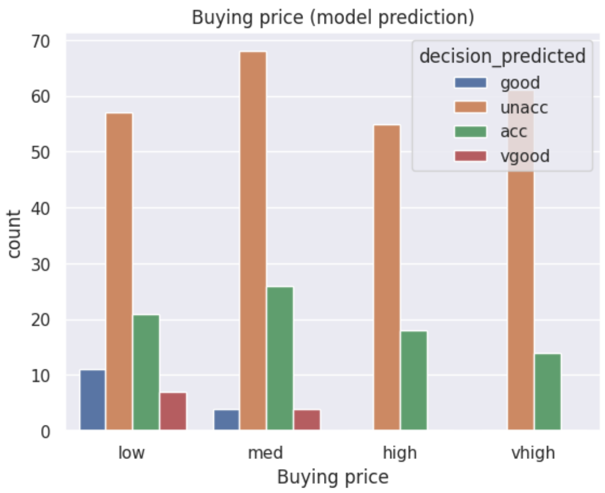

Fig. 10. Acceptabilty of the car based on buying price (model prediction)

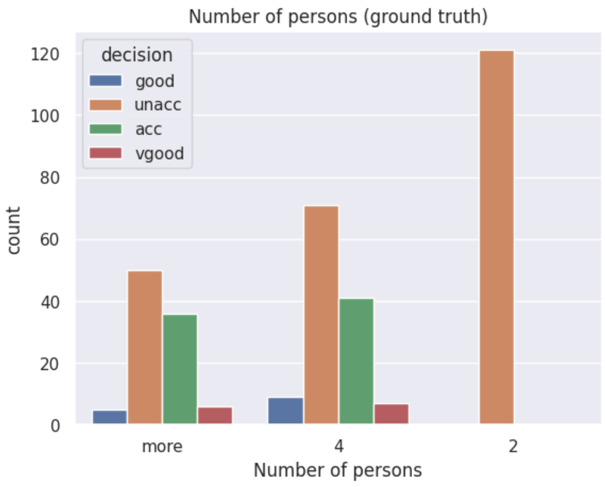

Fig. 11. Acceptance based on passenger capacity (original data)

- From the two figures above and below, it is seen that cars with only 2 passenger capacity are unacceptable. Our general notion says these kinds of cars are primarily race cars, which are really expensive, and for common people with families, a two-seater car is completely useless. We see the largest number of acceptable cars are four-seaters. The trend is the same in both the prediction and original plots.

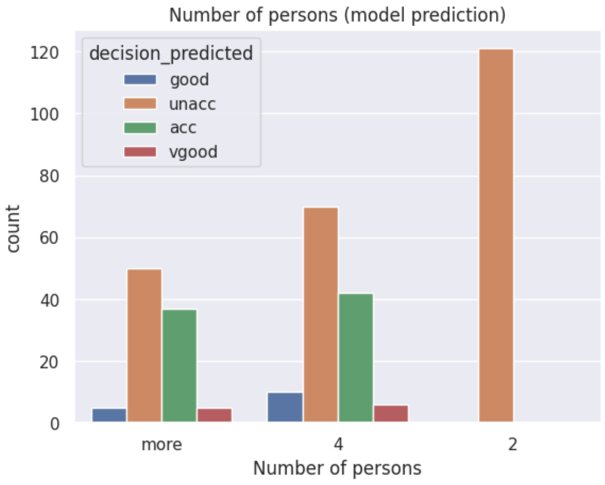

Fig. 12. Acceptance based on passenger capacity (model prediction)

- The plot given below provides information regarding the relation between a car's safety and acceptance. Like the one in the EDA part, a highly unsafe car is unacceptable. This is quite reasonable because no one will prefer to purchase a car whose safety is very low.

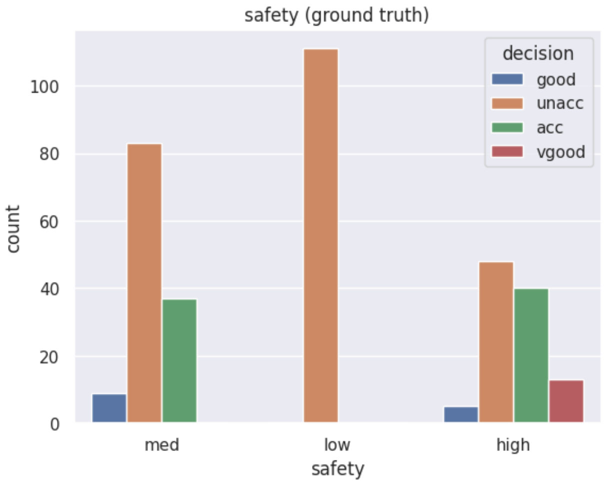

Fig. 13. Safety and acceptance in original data

Further, it is seen that the number of rejected cars is higher for moderately safe cars, too. The acceptability is highest for the safest cars. And the difference between the number of unacceptable and acceptable cars, which are the safest, is also very small. So evidently, people prefer the safest cars over anything else. The true and predicted labels have quite high similarity.

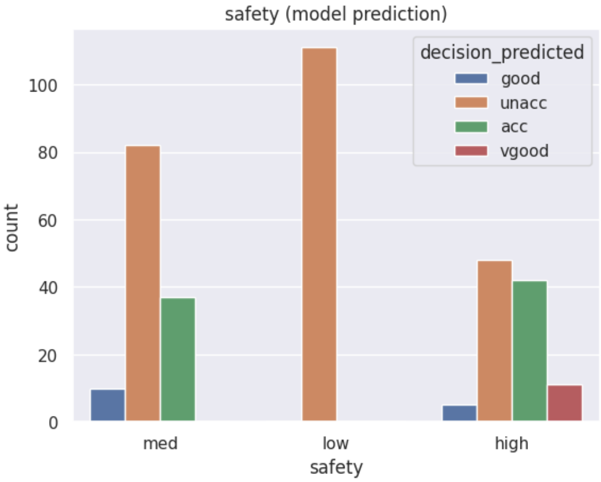

Fig. 14. Safety and acceptance in the model prediction data

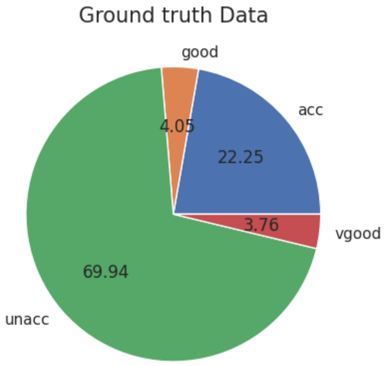

Fig. 15. Pie plot for original class labels in percentage

- From the above plot, it is seen that percentages of class labels are for unacc 69.94%, for acc 22.25%, for good 4.05% and for vgood 3.76% in the labelled test set.

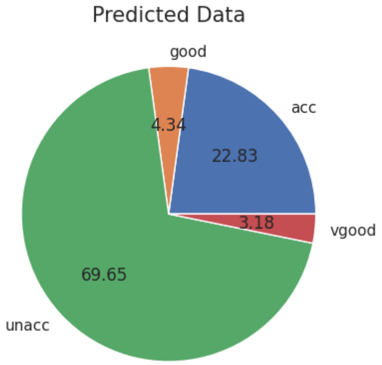

Fig. 16. Pie plot for predicted class labels in percentage

- From above, it is seen that percentages in predicted class labels are for unacc 69.65%, for acc 22.83%, for good 4.34% and for vgood 3.18%. So, for each class, the percentages are almost the same in both the labelled and prediction sets.

So, this visualization really helped to understand the model interpretability a great deal. The similarity between the original and the predicted classes proves that the model is very good in terms of reliability.

V. Conclusions

Decision tree is one of the most popular classification algorithms. It is quite easy to explain when the tree depth is less. Unlike Logistic regression, it can work on both categorical and numerical data. It has the ability to capture complex patterns in the data, which many classification algorithms like Logistic regression, Naive Bayes, and Support vector machines often fail. The classification capacity of this algorithm increases even more when it is used with ensembled techniques like Random forest, where multiple decision trees are used. In the domain of ensembled methods, decision trees are the easiest ones to understand. In this assignment, we successfully demonstrated the applicability of the Decision tree. This assignment allowed us to explore the mathematical details of the Decision tree and other statistical techniques, such as Entropy, Chi-squared and Cramer's V rule for obtaining correlation among the discrete categorical features. The name is consistent with the working process of this algorithm as it extensively works similarly to the construction of trees in Data structures. This dataset allowed us to determine what visualization techniques are useful for categorical data. With these plots, our simple conclusions are expensive, low passenger capacity, unsafe cars, are the ones with the least acceptability.

References

[1] Decison tree: https://www.geeksforgeeks.org/decision-tree/

[2] Evaluation Metrics: https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

[3] Pandas: https://pandas.pydata.org/

[4] Scikit learn Decision tree: https://scikit-learn.org/stable/modules/tree.html

[5] Seaborn: https://seaborn.pydata.org/index.html

Access the original code here