Assignment 5
A mathematical essay on Random Forest

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 a famous ensemble technique called Random Forest 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, Ensemble Methods, Bagging, Random Forest, 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 Random Forest 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 whether a car's acceptance is good, very good, somewhat acceptable or completely unacceptable.
- A Random Forest 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. It uses multiple decision trees for prediction. This makes this model even stronger than the decision trees. Each decision tree works by recursively splitting the randomly chosen sub-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. As multiple trees work together on the random subsets of data, the name Random Forest is quite relevant.
- 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 Random Forest algorithm will be used, just like a decision tree, random forest works very well when the attributes are categorical in nature.
- This paper will demonstrate the data analysis technique relevant to the random forest and 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. Random Forest

Ensemble learning: It is a machine learning technique that combines predictions from multiple models to create a stronger, more accurate predictive model. The basic idea behind ensemble methods is that by combining the predictions of multiple models, the errors of individual models can be mitigated, leading to improved overall performance. Ensemble learning can be applied to both classification and regression problems. There are two kinds of ensemble learning.

Bagging (Bootstrap Aggregating): Bagging involves training identical multiple models on different subsets of the training data. Each model in the ensemble is trained on a random subset of the data with replacement (called bootstrap samples). The final prediction is an average (for regression) or a majority vote (for classification) of the predictions made by individual models. Random Forest is a kind of bagging technique, having a minute difference, which will be discussed later.

Boosting: Boosting works by training a series of weak learners (models that are slightly better than random chance) sequentially. Each model in the sequence focuses on correcting the errors made by the previous ones. Examples of boosting algorithms include AdaBoost, Gradient Boosting, and XGBoost.

The discussion in this paper will be restricted to Random Forest alone.

A. Important Features

- It does not consider all the features in learning the trees. This is helpful when the feature dimension is very large, and the model will be free from the curse of dimensionality.

- 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.
- This algorithm creates trees independently, which can make use of the entire CPU, leading to faster training.
- Because the random forest predicts classes based on the majority vote in the context of classification, it is very stable while training.
- Splitting Criteria are the same as decision trees, such as Information Gain, Gini Index, and Entropy.

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:

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

Training a random forest classifier is essentially same as training multiple decision trees. So whatever mathematical topics are discussed above holds equally for larger random forest model.

C. Bagging

One big limitation of the decision tree is it suffers from high variance. So, a small change in the dataset leads to a completely different decision tree. Here, Bootstrap aggregation comes really handy to mitigate this issue. Training a decision tree on multiple training datasets is often not possible because sometimes accessing those data may be difficult, and there might be data scarcity. So, in bootstrapping, we generate multiple subsets of data from training data where repetition of the dataset is allowed. In order to use bagging, construct a given number of trees, say N, based on N bootstrapped samples. The constructed trees are allowed to grow deeper, so they might suffer from high variance issues, But keeping N also sufficiently large eliminates this problem, as we choose the prediction based on majority voting.

D. Random Forest

It behaves similarly to bagging, but trees work on a randomly chosen subset of features instead of the actual number of features. The main purpose of this kind of training is to keep the trees as uncorrelated as possible, and this reduces variance significantly for both regression and classification.

E. 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.

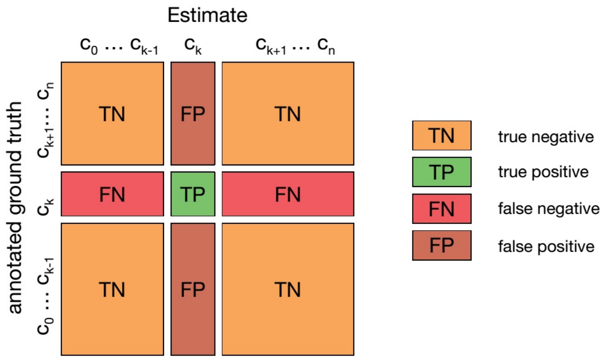

Fig. 1. Confusion matrix for multiclass classification

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.

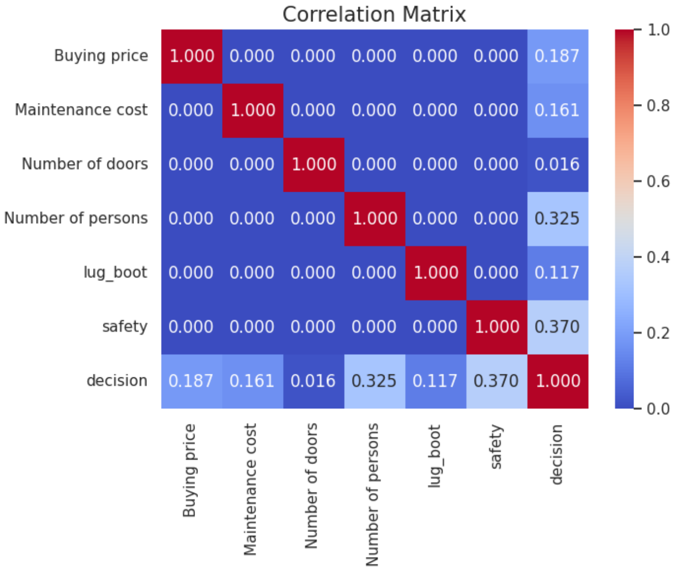

Fig. 2. Correlation plot

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.

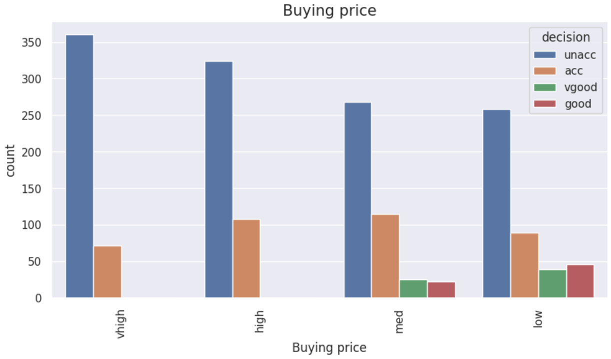

Fig. 3. Buying price and acceptability

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.

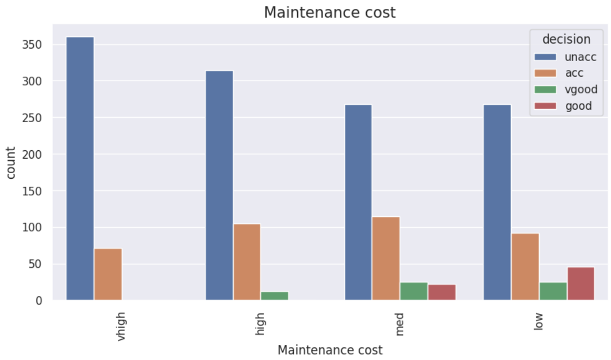

Fig. 4. Acceptabilty with maintenance cost

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

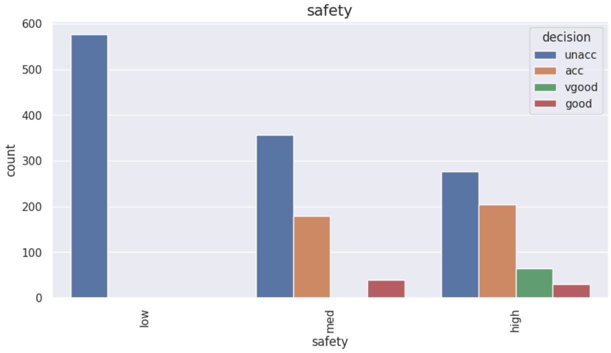

Fig. 5. Acceptance with car safety

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.

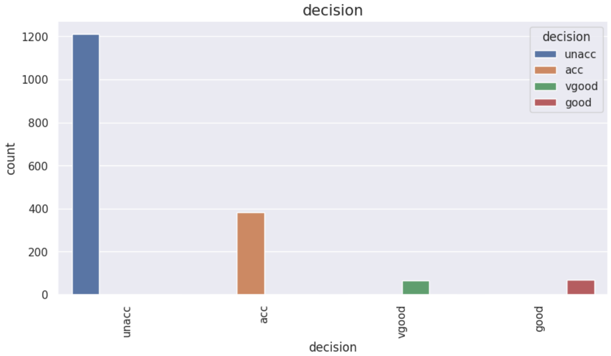

Fig. 6. Absolute count of 4 classes

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

The data will split into training and test sets to validate the model, with the 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 Random forest classifier from sklearn. In order to perform proper hyperparameter tuning, the models are trained with the Grid search cross-validation technique. There were several parameters chosen, such as the number of estimators, the minimum number of leaf nodes, the maximum number of features, etc. With the best hyperparameters, the model achieved fairly high performance. It gave 99% accuracy on the holdout test set.

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	1.00	1.00	1.00
Unacc	1.00	1.00	1.00
Vgood	1.00	0.92	0.96

TABLE II
EVALUATION METRICS

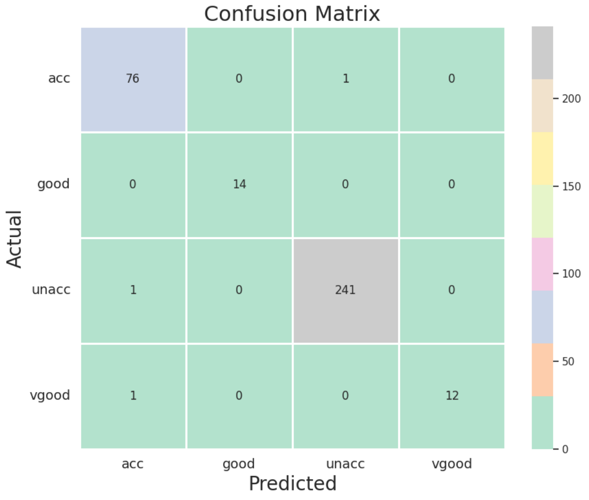

Fig. 7. Confusion Matrix with heatmap

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; there is no other true class for which prediction is good. So precision is 1. 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; there is one true class label acc, which is predicted to be of unacc class. So precision is 0.995. 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. In the estimation, these are rounded off to 1.00. 0.995 is very close to one.
- Similarly, for the vgood class, achieved precision and recalls are 1 and 0.92, respectively.

From the confusion matrix, we get the accuracy as:

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

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

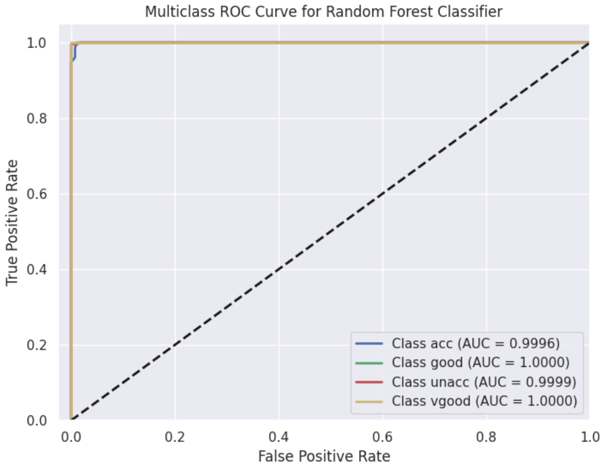

Fig. 8. Receiver operating characteristic curve

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 0.9996 for class Acc, 1.0000 for class Good, 0.9999 For class Unacc and 1.0000 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.

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.

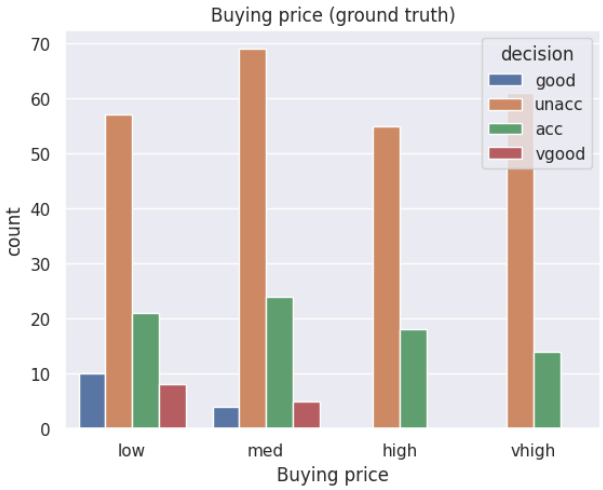

Fig. 9. Acceptabilty of the car based on buying price (original data)

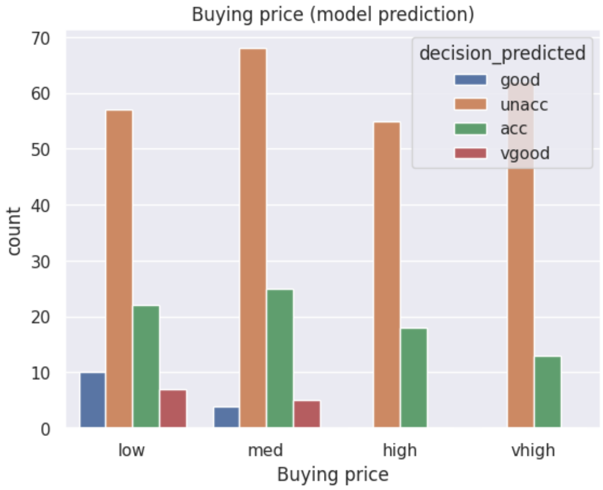

Fig. 10. Acceptabilty of the car based on buying price (model prediction)

- The cars that are costly to maintain are usually less acceptable. The bar plots below also convey the same message.

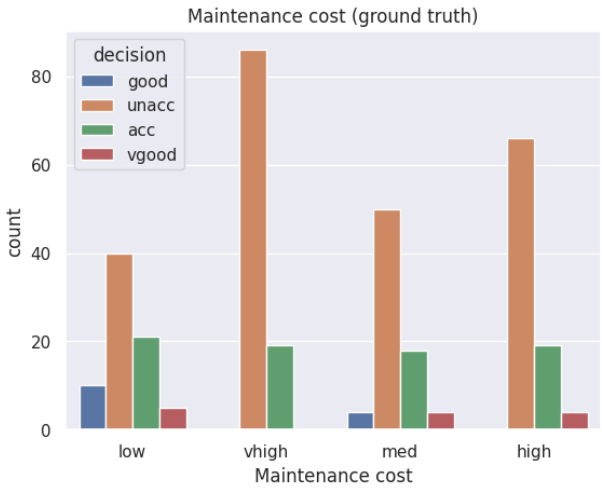

Fig. 11. Acceptance based on passenger capacity (original data)

The bar plot of predicted data is quite similar to the above plot. In both cases, there is no entry of the vgood and good class for the cars that charge the highest for maintenance. This implies people don't consider these kind of cars good or very good in terms of safety.

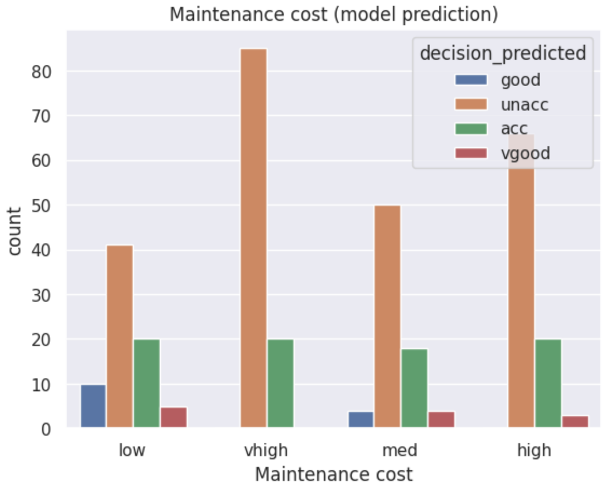

Fig. 12. Acceptance based on passenger capacity (model prediction)

The cars that have very low maintenance costs have the least unacceptance frequency, which is quite relevant. The number of acceptable cars across all the types of low, high, high, and med is more or less the same. The class with the lowest frequency is vgood across all the price ranges of car maintenance. Both ground truth and predicted ones are exactly identical in this case.

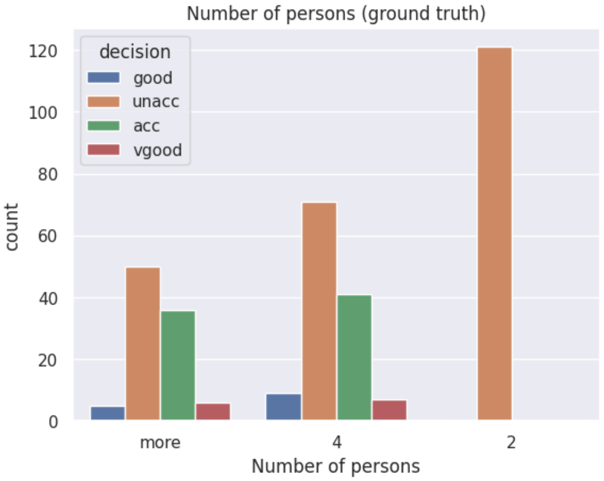

Fig. 13. Acceptance based on passenger capacity (original data)

- 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.

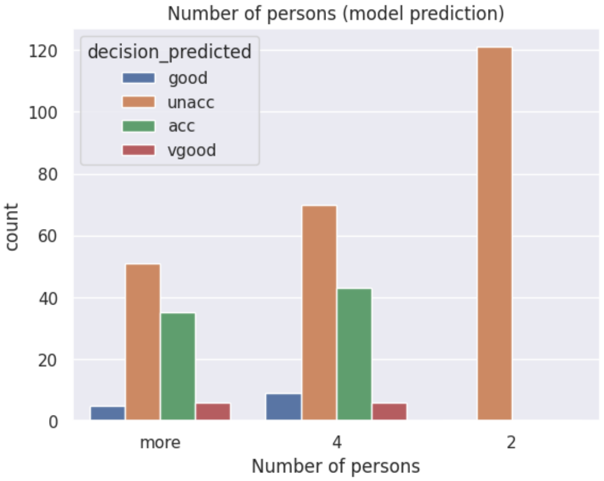

Fig. 14. Acceptance based on passenger capacity (model prediction)

- 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.

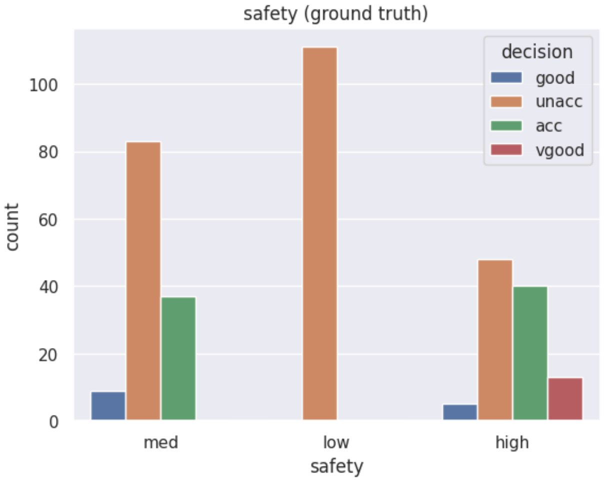

Fig. 15. Safety and acceptance in original data

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.

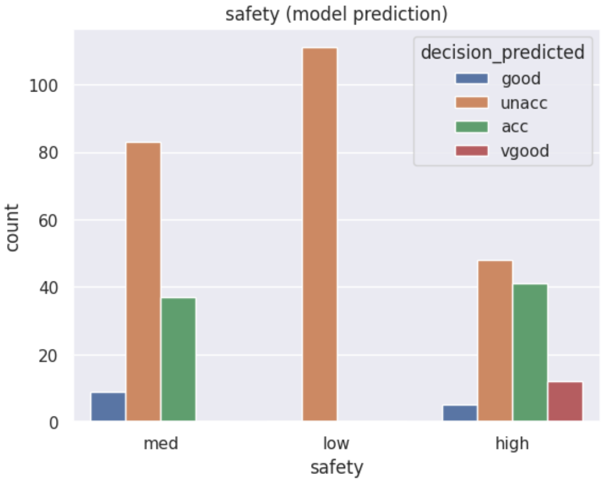

Fig. 16. Safety and acceptance in the model prediction data

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

Random Forest is one of the most popular ensemble algorithms. 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 is way higher than that of conventional decision trees just because of the fact that it uses multiple decision trees. Use of multiple decision trees makes it very robust against the variance of the data. In this assignment, we successfully demonstrated the applicability of the Random forest. This assignment allowed us to explore the mathematical details of the Random forest 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 uses a collection of trees. 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, and unsafe cars are the ones with the least acceptability.

REFERENCES

[1] Random Forest: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

[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 gridsearchcv: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[5] Scikit learn random forest: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

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

Access the original code here