Assignment-5: A Mathematical Essay on Random Forest Classifier

Shania Mitra
Roll Number: CH18B067
Chemical Engineering
IIT Madras, Chennai
ch18b067@smail.iitm.ac.in

***Abstract*—In this study, we estimate the safety of a car based on factors such as buying price, maintenance price, capacity, etc. A random forest classifier is used to model the importance of these factors and predict the safety of cars.**

***Index Terms*—Random Forest, Visualization, Car Evaluation**

I. Introduction

This study empirically analyzes the factors affecting car safety using the Car Evaluation Database. We find that a number of factors, such as maintenance price, purchase price, luggage and seating capacity to significantly affect the safety category of cars.

Random forest is a supervised learning algorithm. The ”forest” it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

In this study, we use Random Forests to model the safety category of individuals based on maintenance price, purchase price, luggage and seating capacity etc.

We begin by gathering, cleaning and preparing the data, following which we perform exploratory analysis. Finally, we build statistical models and perform visualizations to provide quantitative and visual evidence of the relationships observed. In the next section, we highlight the key principles underlying Random Forests. In section 3, we discuss the insight and observations drawn from the data and the models. Finally, in section 4 we outline the salient features of the study and present further avenues of possible investigation.

II. Random Forests

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, \ldots, x_n$ with responses $Y = y_1, \ldots, y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples: For $b = 1, \ldots, B$: 1. Sample, with replacement, n training examples from X, Y; call these X_b, Y_b. 2. Train a classification or regression tree f_b on X_b, Y_b. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x' :

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b\left(x'\right)$$

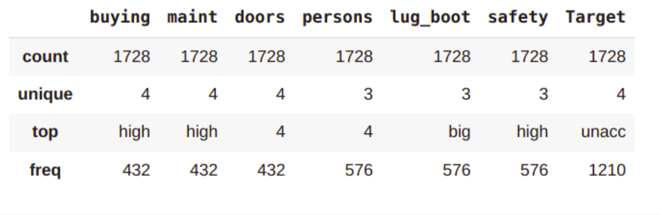
or by taking the majority vote in the case of classification trees. This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets. Additionally, an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees on x ':

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}\left(f_b\left(x'\right) - \hat{f}\right)^2}{B-1}}$$

The number of samples/trees, B, is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set. An optimal number of trees B can be found using cross-validation, or by observing the out-of-bag error: the mean prediction error on each training sample x_i, using only the trees that did not have x_i in their bootstrap sample [15] The training and test error tend to level off after some number of trees have been fit.

III. The problem

In this study, we estimate the safety of a car based on factors such as buying price, maintenance price, capacity, etc.



	buying	maint	doors	persons	lug_boot	safety	Target
count	1728	1728	1728	1728	1728	1728	1728
unique	4	4	4	3	3	3	4
top	high	high	4	4	big	high	unacc
freq	432	432	432	576	576	576	1210

TABLE I
DESCRIPTION OF EACH FEATURE

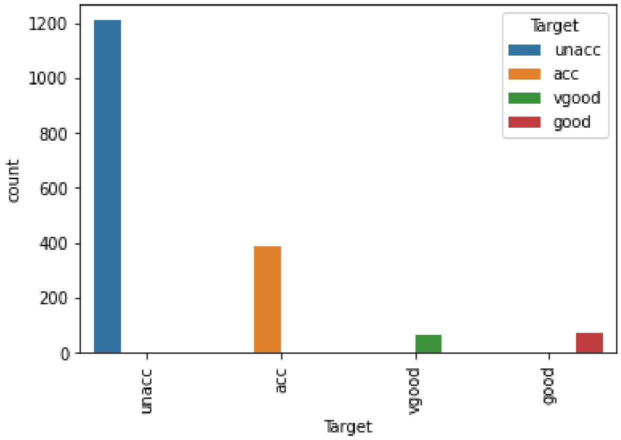


Fig. 1.

A. Data Preparation

The training dataset used in this study consists of 1728 cars and 6 features. Interpretation of the features is as follows:

- Buying: The price of purchase of the car - ’vhigh’ ’high’ ’med’ ’low’
- Maintenance: The cost of maintenance of the car - ’vhigh’ ’high’ ’med’ ’low’
- Persons: Seating capacity of the car - ’2’ ’4’ ’more’
- Doors: The number of doors in the car - ’2’ ’3’ ’4’ ’5more’
- Lug_boot: The boot space of the car - ’small’ ’med’ ’big’
- Safety: ’low’ ’med’ ’high’
- Target: ’unacc’ ’acc’ ’vgood’ ’good’

No missing values are present in the dataset. The aim is to predict the multiclass feature Target.

B. Exploratory Analysis

In this section we look at the trends between various features.

In Table I we have a description of the features and the count, number of unique values and mode in each of them. The dataset has 1728 cars, with no missing values.

In figure 1, we plot the count of each of the target classes and observe that the class “unacc” occurs the most frequently, followed by “acc”.

In figure 2 we plot the count of maintenance cost classes split by the targets. It is observed that for each of the maintenance classes target “unacc” and “acc” occur most frequently. Further, in maintenance class “vhigh” no cars of target “vgood” or “good” occur. Thus, if a car has very high maintenance, the target would most likely not be “vgood” or “good”.

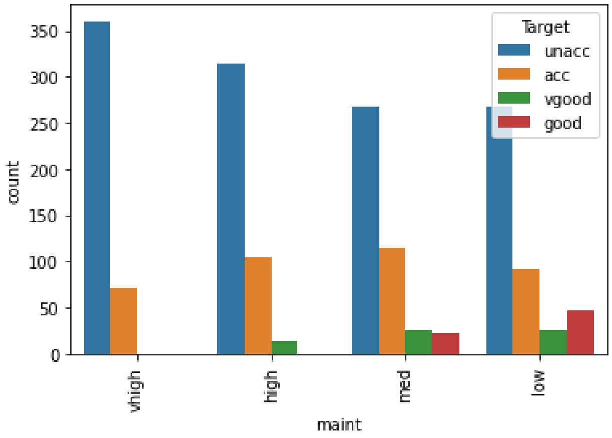


Fig. 2. Count plot of Maintenance Cost Classes split by Target

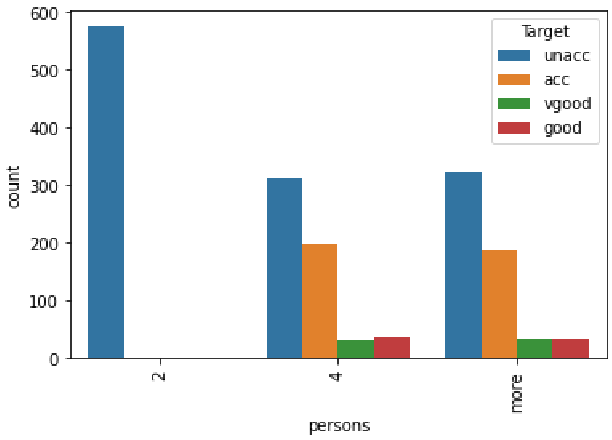


Fig. 3. Count plot of Number of persons split by Target

In figure 3, we plot the count of capacity classes split by the targets. It is observed that for each capacity, target “unacc” occurs most frequently. Further, with cars of capcity 2, only cars of target “unacc” occur. Thus, if a car has a capacity of 2, possibly a luxury or sports car, the target would most likely be “unacc”.

In figure 4, we plot the count of lug_boot classes split by the targets. It is observed that for each capacity, target “unacc” and “acc” occur most frequently. Further, cars with small boot space no samples with target “vgood” occur. Thus, if a car has small bootpsace, such as an SUV, it most likely would not have a target of “vgood”.

In figure 5sr, we plot the count of safety classes split by the targets. It is observed that for each safety, target “unacc” occurs most frequently. Further, cars with low safety have target as “unacc” only, while, cars with medium safety have no samples with target “vgood”. Thus, if a car has a target of

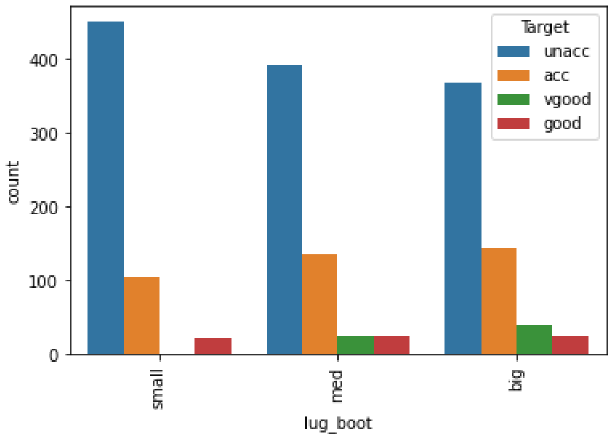


Fig. 4. Count plot of Lug_boot Classes split by Target

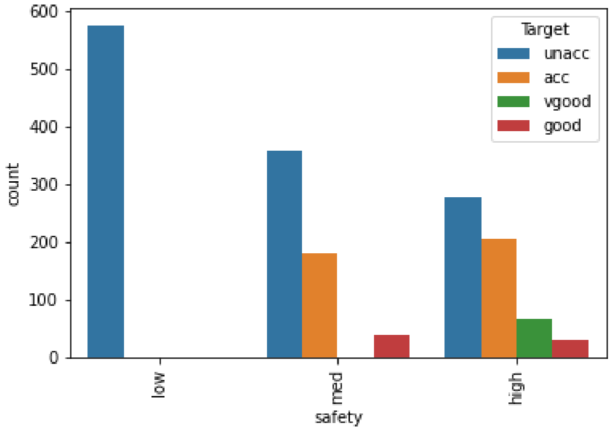


Fig. 5. Count plot of Safety Classes split by Target

vgood, it will most likely have high safety.

It must be pointed out that the joint distribution of the features (i.e., except for Target) amongst themselves is uniform. For example, the count of cars in various buying and maint classes is the same as can be seen in the table below.

```
buying  maint
high    high     108
        low      108
        med      108
        vhigh    108
low     high     108
        low      108
        med      108
        vhigh    108
med     high     108
        low      108
        med      108
        vhigh    108
vhigh   high     108
        low      108
        med      108
        vhigh    108
dtype: int64
```

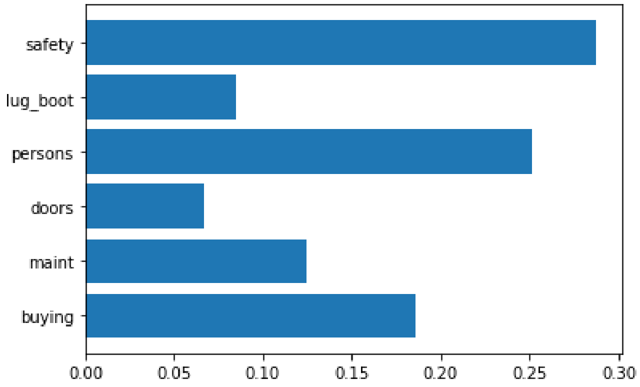


Fig. 6. Feature importances learnt by the Random Forest model

C. Random Forest Model

To train the model and gauge its performance, the dataset was split into train and test sets with an 80-20 split. Following are the metrics achieved by the Random Forest model:

```
              precision    recall  f1-score   support

         acc       0.90      0.94      0.92       193
        good       1.00      0.80      0.89        40
       unacc       0.99      0.98      0.98       599
       vgood       0.85      0.91      0.88        32

    accuracy                           0.96       864
   macro avg       0.94      0.91      0.92       864
weighted avg       0.96      0.96      0.96       864
```

In Figure 6, we observe the importances of the features as learnt by the model. It can be seen that safety and seating capacity are the most important features.

D. Comparison with Decision Tree Model from Assignment-4

Similar to the Random Forest model training, here again the dataset was split into train and test sets with an 80-20 split. Following are the metrics achieved by the Decision Tree model:

```
              precision    recall  f1-score   support

         acc       0.95      0.92      0.93       193
        good       0.97      0.72      0.83        40
       unacc       0.98      0.98      0.98       599
       vgood       0.71      0.91      0.79        32

    accuracy                           0.96       864
   macro avg       0.90      0.88      0.88       864
weighted avg       0.96      0.96      0.96       864
```

Thus, we observe that the test set has minimum number of samples for classes vgood and good and acheives an f1 score below average for these, while the model performs very well on classes “unacc” and “acc” due to large number of samples.

From the metric tables of the two models it can be observed that on class “acc”, the decision tree (DC) has acheieved a higher f1-score as compared to Random Forest (RF). On class “unacc” they acheive the same score while on classes “good” and “vgood”, having lesser number of samples, RF performs better. This tells us that the RF model is better at handling cases with extreme class imbalance.

IV. Conclusions

In this study, we observed that cars with very high maintenance cost, small capacity, small bootspace and low safety are most likely to have target as “unacc”. Further, cars with target as “vgood” are most likely to have high safety. In the future, techniques such as SMOTE to handle class imbalance could be implemented.

References

[1] Random Forest. (2021, November 4). In Wikipedia. https://en.wikipedia.org/wiki/Random_forest

[2] Donges, N. (2021, September 17). A Complete Guide to the Random Forest Algorithm. Built In. https://builtin.com/data-science/random-forest-algorithm

[3] Z. (2021, September 26). Random Forest Explained - Towards Data Science. Medium. https://towardsdatascience.com/random-forest-explained-7eae084f3ebe