A mathematical essay of Random Forests

Devansh Sanghvi
Department Of Biotechnology
Indian Institute of Technology, Madras
Chennai, India
be19b002@smail.iitm.ac.in

***Abstract*—This paper aims to develop an understanding of the mathematical properties of random forests and showcase its application in classifying a car based on its safety.**

I. Introduction

Random Forests is one of the most important tools used for classification and prediction. Once we receive data with some independent variables and a target variable which we want to determine for a given data point, we use statistical tools to develop a relationship between the target and the independent variables to decide on mathematical rules to assign a given data point to a class. The random forest classifier uses multiple decision trees for the classification.

Random Forests have applications spanning multiple fields. Like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The figure below depicts the same.

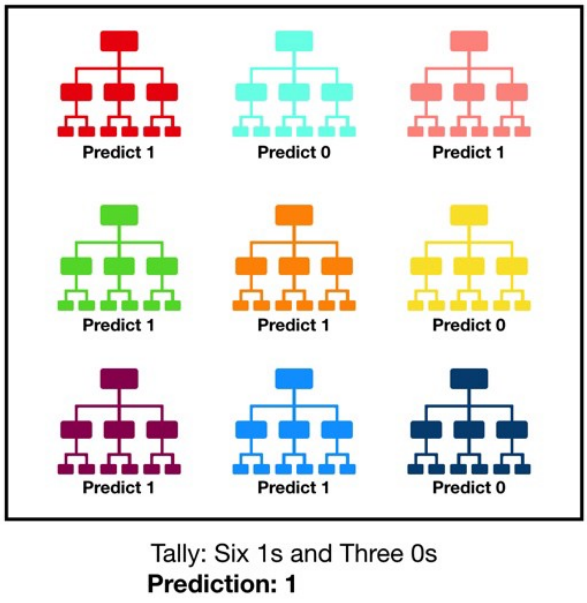

Fig. 1. Basic intuition of random forests

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds.A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models, and hence random forests are such a powerful method. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction.

So the prerequisites for random forest to perform well are:

- There needs to be some actual signal in our features so that models built using those features do better than random guessing.
- The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

So how does random forest ensure that the behavior of each individual tree is not too correlated with the behavior of any of the other trees in the model? It uses the following two methods:

- **Bagging**: Decisions trees are very sensitive to the data they are trained on — small changes to the training set can result in significantly different tree structures. Random forest takes advantage of this by allowing each individual tree to randomly sample from the data set with replacement, resulting in different trees. This process is known as bagging.
- **Feature Randomness**: In a normal decision tree, when it is time to split a node, we consider every possible feature and pick the one that produces the most separation between the observations in the left node vs. those in the right node. In contrast, each tree in a random forest can pick only from a random subset of features. This forces even more variation amongst the trees in the model and ultimately results in lower correlation across trees and more diversification.

This paper will serve as a part of the requirements to be fulfilled for receiving credits in the course **EE4708: Data Analytics Laboratory**.

II. Random Forests

As mentioned in the section above, random forests use multiple trees made from random features and random data using bagging and feature randomness. In this section we will discuss some advantages, disadvantages, and the metrics of evaluation for random forests:

- Advantages
  1) Random Forests reduces the over fitting problem in decision trees and also reduces the variance and therefore improves the accuracy.
  2) Random Forest works well with both categorical and continuous variables.
  3) Random Forest can automatically handle missing values.
  4) No feature scaling is (standardization and normalization) required in case of Random Forest as it uses rule based approach instead of distance calculation.
  5) Random Forest is usually robust to outliers and can handle them automatically.
  6) Random Forest handles non-linear parameters efficiently.
- Disadvantages:
  1) Random Forest creates a lot of trees and combines their outputs. To do so, this algorithm requires much more computational power and resources. On the other hand decision tree is simple and does not require so much computational resources.
  2) Longer Training Period: Random Forest require much more time to train as compared to decision trees as it generates a lot of trees (instead of one tree in case of decision tree) and makes decision on the majority of votes.
- Metrics of evaluation: As for many other machine learning algorithms, we can extract the classification report for our classification. The main metrics from the classification report that are of our interest are:
  1) Precision: How many values predicted to be in a certain class are in that class
  2) Recall: How many values in each class were given the correct label
  3) F1-score: weighted average of precision and recall
  4) Accuracy: This score measures how many labels the model got right out of the total number of predictions. Accuracy is not a great measure of classifier performance when the classes are imbalanced. We need more information to understand how well the model really performed.

  The metric which you give the most importance to depends on your interest in false positives/ false negatives.

III. Pre-processing and exploratory analysis on the data

In this section, we will first pre-process the data to extract the important features in the data provided, then we will visualize the data to understand the effects of certain features and modify the data-set to optimize the results from our random forests model.

A. Pre-processing the data

The initial data-set has 1727 samples and 6 independent features, namely: the cost of buying the car, the cost for maintenance, number of doors in the car, number of persons that can sit in the car, how large is the luggage boot and predicted safety of the car. . Pre-processing the data-set includes (Note that the changes are made in both the training and testing data-sets):

1) Missing value assessment: No features in our data set have missing values, hence we can skip this part of pre-processing.
2) Additional Variables: No features seem to depend on each other and hence, it is safe to include every feature in our model.

B. Visualization

After preprocessing the data, we visualize the data, to better understand the correlation between different features:

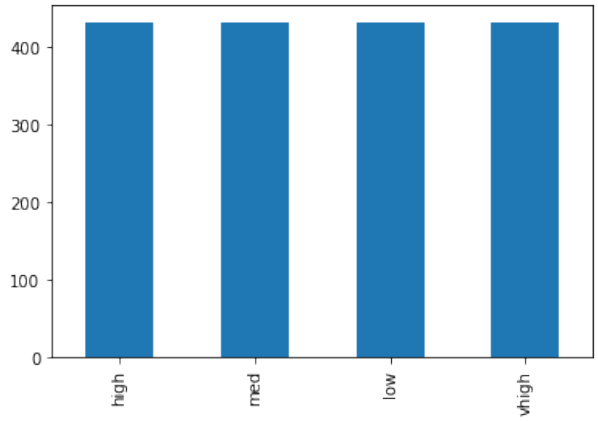

Fig. 2. Count plot of the 'buying' column in the data set

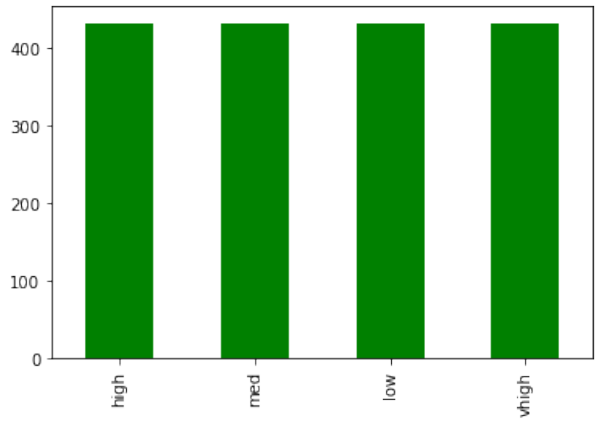

Fig. 3. Count plot of the 'maintenance' column in the data set

1) Exploration of the countplots: The independent variables are almost unskewed, with an almost equal distribution of the counts of all the independent variables. The dependent variable seems to show a large number of 'unacc' values, meaning that most of the cars are unacceptable with respect to their features.
2) Countplot of the luggage boot size per target: As the luggage boot size increases, the acceptance rate for the

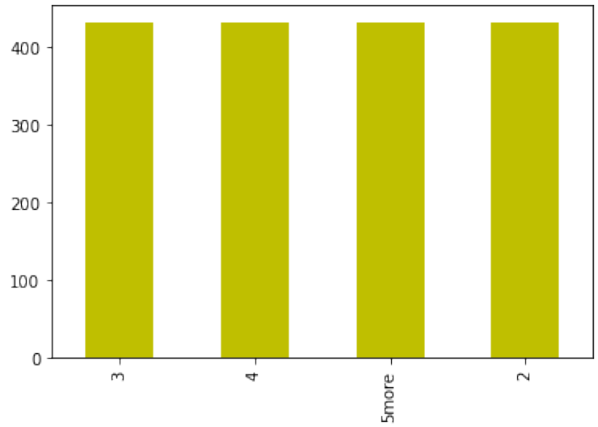

Fig. 4. Count plot of the 'doors' column in the data set

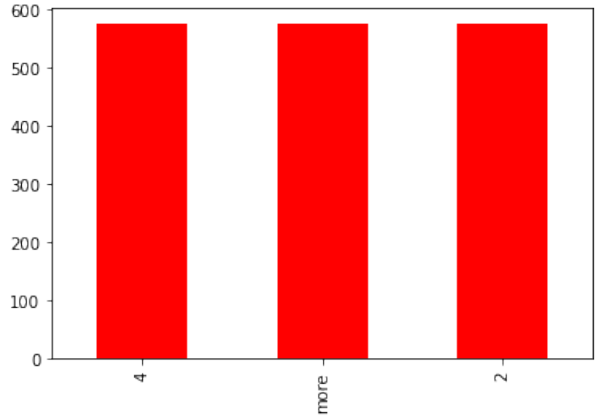

Fig. 5. Count plot of the 'persons' column in the data set

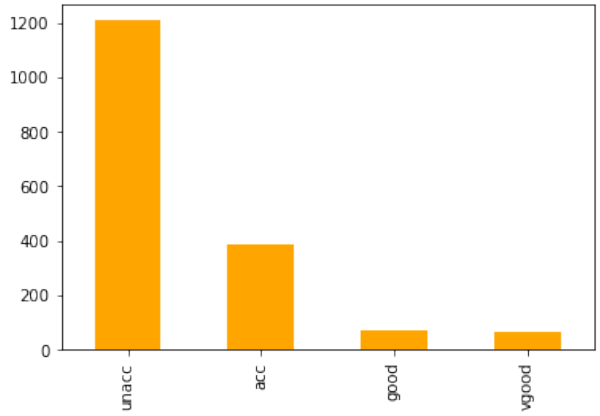

Fig. 6. Count plot of the 'target column in the data set

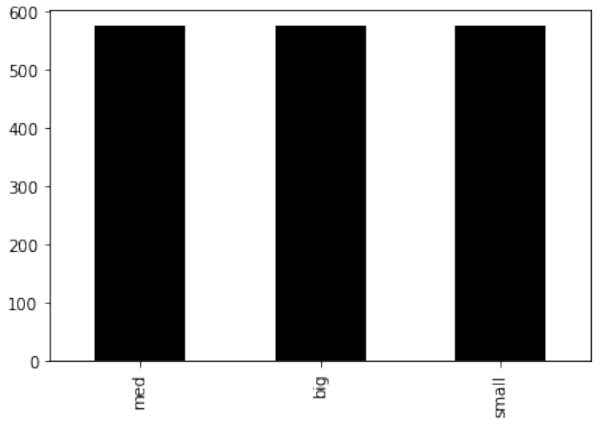

Fig. 7. Count plot of the 'luggage boot' column in the data set

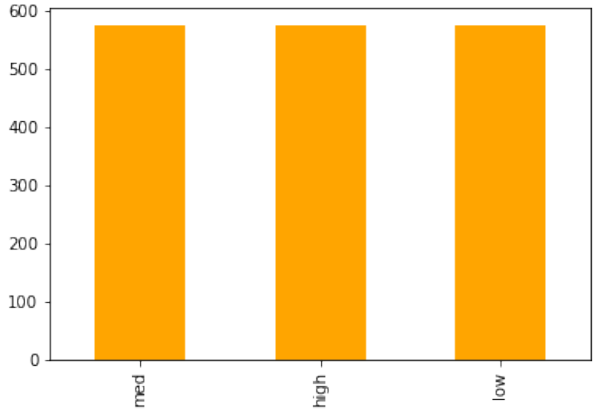

Fig. 8. Count plot of the 'safety' column in the data set

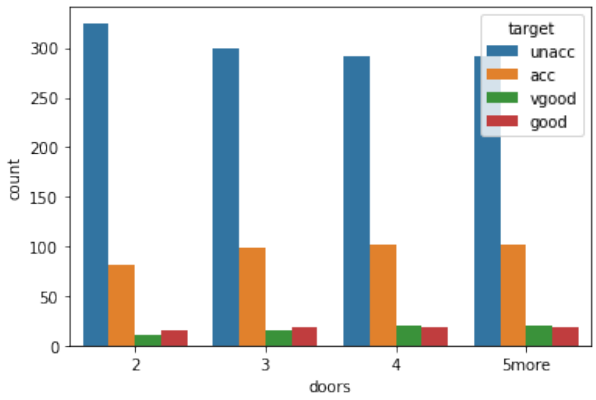

Fig. 9. Count plot of the number of doors per target

cars increases. The number of cars that are unacceptable are similar in number, but decrease as the luggage boot size increases.

3) Countplot of the luggage boot size per target: All cars with a low predicted safety rating are unacceptable and should be rejected. More the predicted safety, higher the acceptance rate amongst those cars.
4) Trend amongst the cost for unacceptable cars: Surprisingly, most cars which are unacceptable cost really large amounts of money, and the acceptance rate increases as the cost decreases.

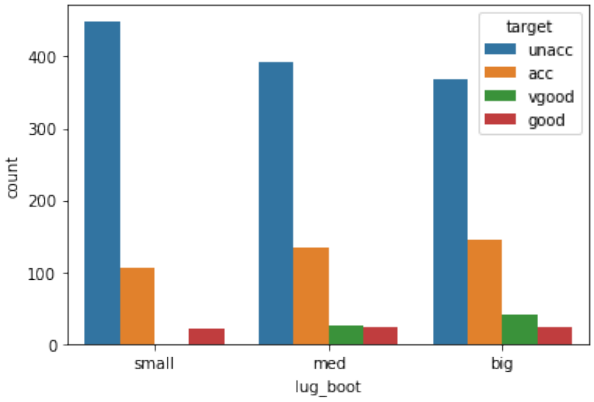

Fig. 10. Count plot of the luggage boot size per target

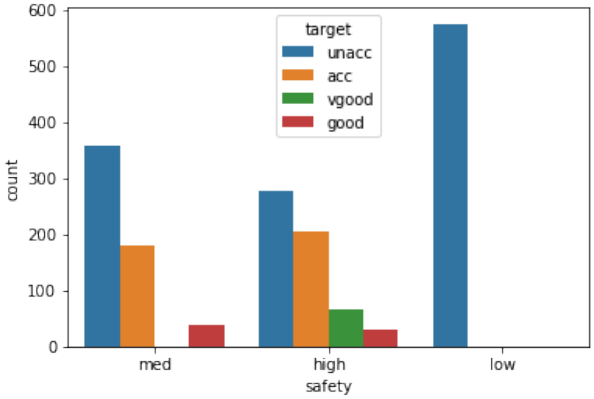

Fig. 11. Plot to compare the predicted safety levels and the target values

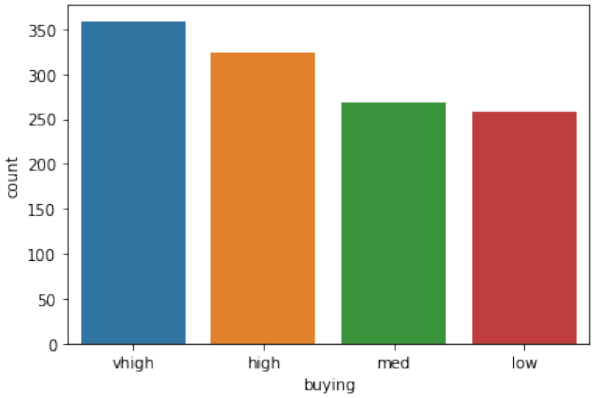

Fig. 12. Count plot of the buying column, when the target column is set to 'unacc'

IV. The Random Forest Model

After preprocessing the data, we move to the heart of the problem: the prediction of whether an unseen car (not in the training data-set) should be accepted. Here are the steps for the same:

- Appropriate Data type: Before applying a decision trees model to our data, we will analyze the data types for all variables, and change them, when necessary. Initially, all variables are of the categorical category. The scikit-learn package used to build the model cannot use categorical variables, and hence we will have to undergo a conversion process. Each variable follow a rank and order and thus, we perform an ordinal encoding on the data to maintain the order. For instance, the size of the luggage boot has three categories: small, medium, and large which has an order.
- Random Forest model and results After the conversion of the data types, we run the random forests model using the simple test/ train model and evaluate our results using the classification report. The classification report can be viewed below:

	precision	recall	f1-score	support
acc	0.94	0.99	0.97	121
good	1.00	0.90	0.95	20
unacc	1.00	0.99	0.99	357
vgood	1.00	0.95	0.98	21
accuracy			0.98	519
macro avg	0.99	0.96	0.97	519
weighted avg	0.99	0.98	0.98	519

Fig. 13. Classification report for the predicted random forest

As shown by the results, random forests performs the classification excellently and delivers great results. The overall accuracy is 98%.

V. Conclusion

The random forest model was succesfully demonstrated, both visually and numerically. While the accuracy of the model is good enough, we could perform feature selection using feature scores.

References

[1] "Car Evaluation Data Set", archive.ics.uci.edu, 1990 [Online]. Available: **Link**.