DAL Assignment 5: A Mathematical Essay On Random Forest

C Vamshi Krishna Reddy
CH20B112
EE5708: Data Analytics Laboratory
IIT Madras, TamilNadu, India
ch20b112@smail.iitm.ac.in

***Abstract*—This document serves as a model for showcasing the application of the Random Forest to build a model that helps us to classify a car into any one of the following classes: very good, good, acceptable, and unacceptable based on six different parameters like buying price, price of the maintenance, number of doors, capacity in terms of person to carry, size of luggage boot and the estimated value of the car.**

***Index Terms*—Classification, Random Forest, Statistical Modeling, Supervised Learning**

I. Introduction

We all know that classification problems are one of the primary tasks solved with the help of Machine learning apart from regression. These algorithms create patterns between input and output variables using already existing data and classify the target variable based on the new input data provided. Broadly speaking, there are 3 types of machine learning algorithms: supervised, unsupervised, and reinforcement learning. In this paper, we are using an Ensemble of Decision trees which is nothing but a Random Forest, which comes under the supervised category.

A Random Forest model is a supervised learning technique in which algorithms are taught from the labels associated with the data. It is a predictive model that uses a set of binary rules to determine the value of the target variable. Classification models are produced by decision trees in the shape of trees. Employing the potential outcomes of each attribute as a branch of the tree aids in understanding the decision hierarchy and relationships between the qualities. Apart from these advantages, it is also important to note that these decision trees are prone to overfitting and they usually can not reach the level of accuracy provided by alternative classification methods also any small change in the data can significantly affect the tree structure for predictions. Decision-tree biases are also eliminated by random forests by averaging out all of their predictions. Therefore, the issue of overfitting is not present. The random forest classifier is also used to choose features. It entails choosing the most significant features from the training dataset's available features.

This paper aims to use Random Forest in explaining the effect of buying price, price of maintenance, number of doors, etc., on the class of the car which can help us in choosing whether to buy a car or not.

The paper provides a detailed overview of Random Forest by solving a real-world problem and gives insights into the relation between variables like buying price, price of maintenance, number of doors, capacity in terms of persons to carry, size of luggage boot, and the estimated safety value of the car and the class of the car.

II. Random Forest

Classification is a widely employed application in machine learning, especially in supervised settings. It essentially helps us understand how certain independent factors determine which class it belongs to. In simpler terms, it's like a tool used in machine learning to make classifications. The core idea is to teach algorithms how different independent variables are linked to an outcome or dependent variable. Once the model grasps this relationship, it becomes capable of predicting what might happen when faced with new, unexpected data or completing missing parts in existing data.

A Random forest, as the name suggests is an ensemble of decision trees that are constructed on 'm' randomly selected features out of 'n' features. Here each decision tree is composed of mainly three sections: root node, branches, and leaves. The root node is the position of the first split. Branches are the subsequent questions that classify deeper and leaves are the end nodes of a decision tree where the prediction process is completed. Random forest is a bit difficult to interpret compared to decision trees but it helps us in preventing overfitting which might occur in deep decision trees. Random forests algorithm also ranks the importance of various features in a regression or classification problem and thus can be used for the feature selection process. In machine learning Random forests are classified into two subgroups: Classification and Regression Random forest.

•**Random Forest Classifier:** Here the Decision trees classifies a target variable and the random forest model chooses the class with maximum votes.

•**Random Forest Regressor:** Here the decision trees predicts what is likely to happen, given previous behavior or trends and the model selects the best solution by taking the mean of the individual output from trees.

A. Cost Function

A random forest is an ensemble of decision trees and each split in decision trees is based on a local metric like information gain/entropy or Gini index and involves a greedy search instead of any Cost function. In fact, even when a global training metric is defined say, likelihood, each step in training is still evaluated based on these local metrics. The impurity in the given data set is known as entropy. Information gain is the term used to describe a decrease in entropy. The average entropy after splitting and the entropy prior to splitting the data set are compared to determine information gain. The splitting attribute at the node is determined by the attribute with the biggest information benefit. The Gini index calculates the probability that a certain variable would be misclassified if it were randomly picked. The Gini index degree ranges from 0 to 1, with 0 indicating that all elements belong to one class and 1 indicating that the elements are randomly distributed among the classes. At that node, the attribute with the lowest gini index for the supplied data is chosen as the splitting attribute.

B. Evaluation Model

In classification models like Random Forest, confusion matrix, and f1 score are used to evaluate the performance of the model. Here since our problem is a Logistic regression which has 4 output classes. We have a confusion matrix of 4 X 4 dimension, And F1-Score is defined as the harmonic mean of recall and precision values of the classification problem. The formula for F1-Score is as follows:

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (1)$$

III. The Problem

The objective is to develop a model to help the person in deciding whether to buy a car or not based on factors like buying price, price of maintenance, the number of doors, capacity in terms of persons to carry, size of luggage boot, and the estimated safety value of the car as the input features. The process initiates with data cleaning and then exploratory data analysis where we predict the 4 possible classes for cars: unacceptable, acceptable, good, and very good.

A. Data Cleaning

The given dataset contains around 1727 rows and 7 columns. Upon looking at the information of data we see that there are no missing values or nulls in the data set. But it is always a good idea to check if nulls are stored in some other format. Even after exploring these variables individually, we can see that there are no nulls in other forms too. So it is safe to say that our dataset is free from nulls or missing values and do not require any further steps of dropping rows or filling up missing data with central tendencies etc.

B. Exploratory Data Analysis and Visualisations

The provided dataset includes car information like maintenance and buying price, the capacity of the car, etc. These details of the car will be used to train a model that can classify a car as unacceptable (or) acceptable (or) good (or) very good, based on the "ground truth" i.e known data. Here we can assume that it would be acceptable to buy a car if it belongs to either acceptable, good, or very good and reject if it belongs to an unacceptable class.

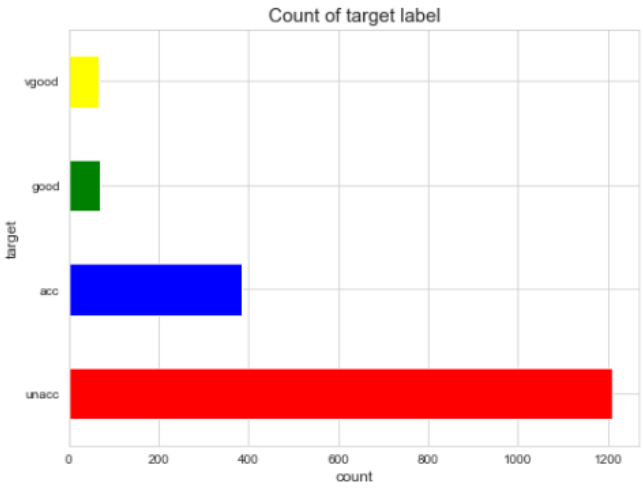

Fig. 1. Count of Target label.

Figure 1 shows a horizontal bar graph plot that tells the count of cars which belonged to four output labels. We can see that count of unacceptable cars is too high compared to remaining three classes count.

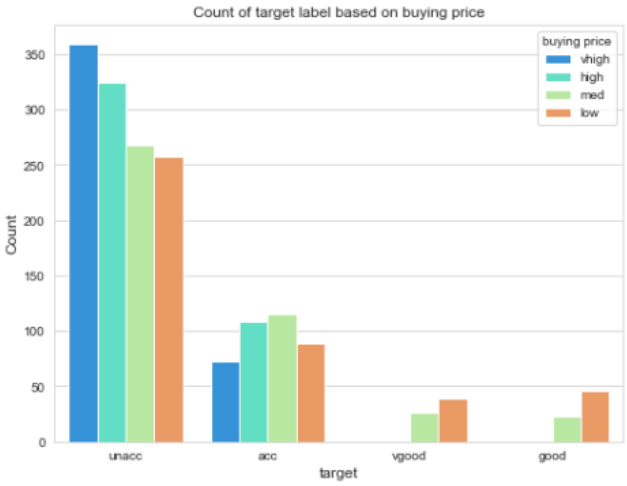

Fig. 2. Count of target labels based on buying price.

From Figure 2, we can see that it is not acceptable to buy a car if its price is too high or high i.e price of a car is inversely affects the probability of buying a car.

From Figure 3 we can see that if the maintenance price of a car is too high then the price of a car inversely affects the probability of buying a car and the count of unacceptable cars are also in good number.

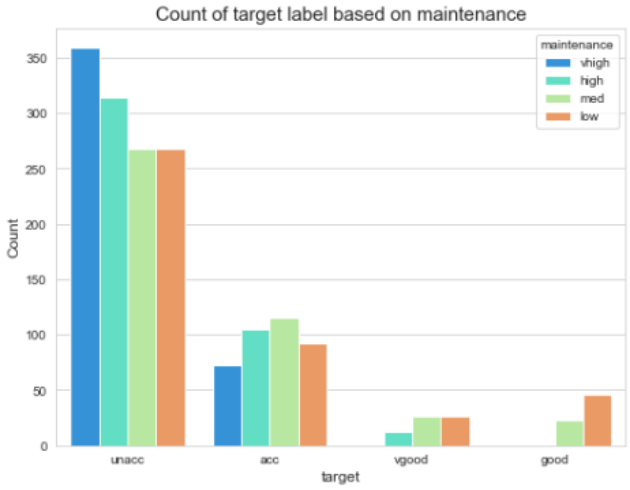

Fig. 3. Count of target labels based on the buying price of maintenance.

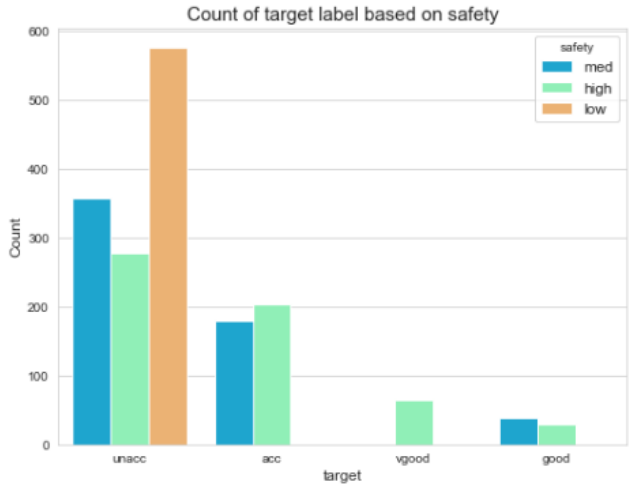

Fig. 4. Count of target labels based on safety of car.

Fig. 5. Count of target labels based on no of capacity of persons.

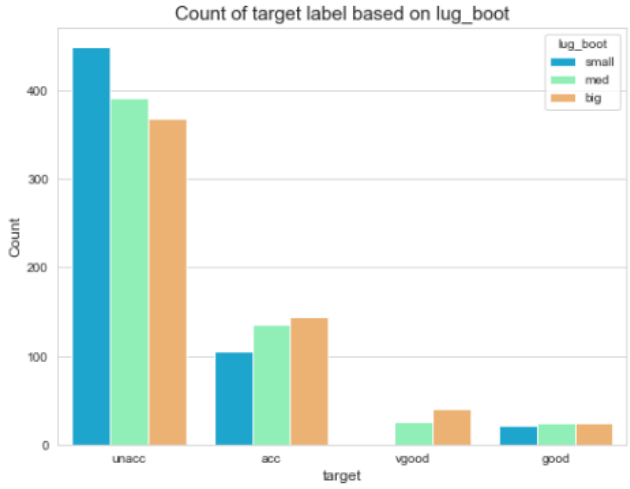

Fig. 6. Count of target labels based on no of lugboot.

Figure 4 gives idea of the effect of safety on the car's target label. We see that cars with low safety are more likely to be unacceptable to buy compared to others.

Fig.5 helps us to conclude that most of the cars that have a capacity of 2 persons are unacceptable to buy.

Fig.6 shows us that the size of the luggage boot directly affects the chance of acceptability as the cars with small luggage boots have a lesser acceptance rate.

C. Model

Now a Random Forest Classifier model is developed on this data set. This data set is first split into 2 data sets x and y. x comprises columns buying, maint, doors, persons, lugboot and safety and y comprises of the Target. We have decided to split these data sets using a train test split with a test size of 0.33 and a random state of 42. To train this model we need to encode the classes in x dataset. As all the columns in x are categorical I used Ordinal Encoder from the category encoders package. Now a model is developed by fitting X train and y train using a Random Forest classifier from sklearn. This trained model is tested by fitting the predicted classes and comparing them with the known classes of the Target column. Using accuracy score from sklearn.metrics we can see that our classification model has an accuracy of 0.92 on the test data set.

There are hyper-parameters for random forest algorithms that must be set before training. Some of the hyper-parameters are max-depth, max features, criterion, and min samples leaf. Each decision tree in the ensemble that makes up the random forest method is built of a data sample taken from a training set with a replacement known as the bootstrap sample. The dataset is subsequently randomized by feature bagging, increasing dataset diversity and decreasing decision tree correlation. The prediction will be determined differently depending on the type of issue. The individual decision trees will be averaged for the regression job, and for the classification task, the predicted class will be determined by a majority vote, or the most common categorical variable. The prediction is then finalized

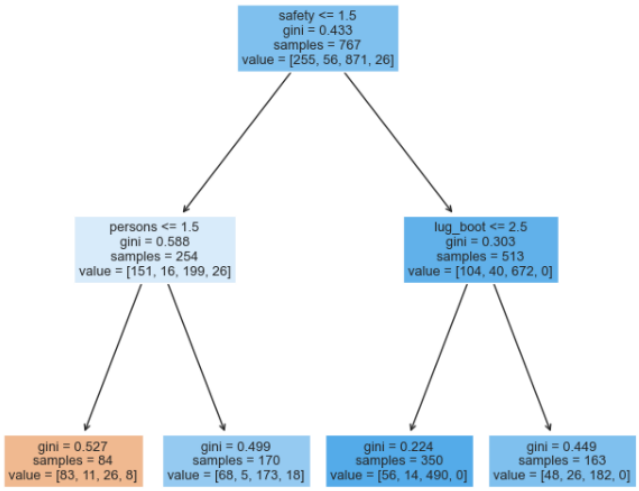

Fig. 7. Random Decision Tree taken from estimators

by cross-validation using the out-of-bag sample. Using the Randomized search CV hyperparameter tuning method, we found the hyperparameters. The best hyperparameters are:

```
RandomForestClassifier
RandomForestClassifier(max_depth=8, max_features=5, min_samples_leaf=6,
                       oob_score=True)
```

Fig. 8. Tuned Hyperparameters.

The classification report of the Random Forest Classifier with the hyperparameters which were tuned is given below. We achieved an accuracy of **0.93** with the tuned hyperparameters.

	precision	recall	f1-score	support
acc	0.97	0.83	0.89	138
good	0.47	0.57	0.52	14
unacc	0.96	1.00	0.98	347
vgood	0.57	0.65	0.60	20
accuracy			0.93	519
macro avg	0.74	0.76	0.75	519
weighted avg	0.93	0.93	0.93	519

Fig. 9. Classification Report.

D. Insights and Observations

From the heat map of correlation, we can see that there is a positive correlation between safety, the capacity of the car, the size of the luggage, and the Target class i.e car with more safety (or) more capacity (or) larger size of the luggage is more likely to belong to acceptable classes (acc, good, vgood). We can also observe a negative correlation between the Buying price, maintenance price, and the Target class i.e. Cars with a larger buying price or maintenance cost are more likely to be in the unacceptable class.

Fig. 10. Heap map of correlation.

IV. Conclusion

With an accuracy of **0.93 or 93 percent**, which is 10 percent higher than the accuracy attained by the decision tree model (85.4 percent), our Random forest model categorizes a car into one of the four classes: very good, good, acceptable, and unacceptable. It does this by taking into account the six attributes, buying price, maintenance cost, number of doors, capacity in terms of people to carry, capacity of the luggage boot, and estimated safety value provided in the data set. The results demonstrate that for car buyers, safety comes first. If a customer thought a car was dangerous, they wouldn't buy it. The data set reveals that customers do not buy cars with less than 2 seats or more than 4 seats, therefore the number of passengers it can hold also plays a significant part in picking a car. The cost of upkeep is also taken into consideration, and people tend to dislike cars with high maintenance prices. The purchasing price also impacts whether it is financially feasible to purchase the car. As the majority of persons in the data set rejected cars with modest capacity, the final feature, luggage capacity, may also deter buyers. The classification reports and confusion matrix both support the reliability of our model.

References

[1] Bishop, Christopher M., Pattern Recognition and Machine Learning, 2006.
[2] https://seaborn.pydata.org/tutorial/categorical.html.
[3] Aurelien Geron., Hands on Machine Learning with Scikit-Learn and Tensorflow, 2019.
[4] https://www.geeksforgeeks.org/random-forest-regression-in-python/.
[5] https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/.
[6] https://towardsdatascience.com/the-f1-score-bec2bbc38aa6.