Assignment 5: Theory of Random Forest Classifier

Shashwat Patel
Metallurgical and Materials Engineering
Indian Institute of Technology,Madras
mm19b053@smail.iitm.ac.in

***Abstract*—Random Forest is a supervised machine learning algorithm that is constructed from multiple decision trees. It is used for both regression as well as classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to a problem. In this paper, we explain the theory behind the random forest classifier and try to understand what parameters are most important while classifying a car based on different parameters.**

***Index Terms*—Random Forest, Ensemble, Bagging, Entropy, Gini Index, Information Gain**

I. Introduction

Cars have now become one of the most important part of our daily lives. The buyer gets so many different choices in cars as there are many different manufacturers. This choice depends mainly on the price, safety, how luxurious and spacious the car is. These parameters vary based on type, model, and manufacturer of the car. The Car Evaluation Dataset was derived from a simple hierarchical decision model. [1]

Random Forest is machine learning algorithm that is used for both regression and classification problems. It is an ensemble method that consists of many decision trees [2]. The random forest algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the mean of the output from various trees. Increasing the number of trees increases the precision of the outcome and doesn't lead to over-fitting.

This algorithm has lot of applications [2]. It is a preferred algorithm over others as it reduces time spent on data management and pre-processing tasks. It is used to evaluate customers with high credit risk, to detect fraud. It is used in computational biology for gene classification, estimating responses to specific medications etc. In e-commerce, it can be used to create recommendation engines.

Buying a car is a very important responsibility. It is important to understand the true financial responsibility that comes when you own a car. The "car_evaluation.csv" dataset is used to identify how parameters like price, price of maintenance, number of doors, capacity, size of luggage boot and safety affect the choice of a buyer while buying a car.

This paper majorly deals with the theory of random forest classifier and the mathematics behind it. We try to understand what kind of parameters are most important while choosing a car to buy using the random forest classifier algorithm. The random forest model is trained on the dataset "car_evaluation.csv".

II. Random Forest

Random forest is one of the most versatile machine learning algorithm. It is a tree based machine learning algorithm which involves building several number of decision trees and then combining their output to improve generalization ability of the model. This method of combining trees is known as ensemble method. Ensembling is nothing but a combination individual trees to produce a stronger model. It is used to solve both regression and classification problems. [3]

To understand the working of random trees, it is necessary to understand the decision tree algorithm. The basic structure of decision tree algorithm looks like:

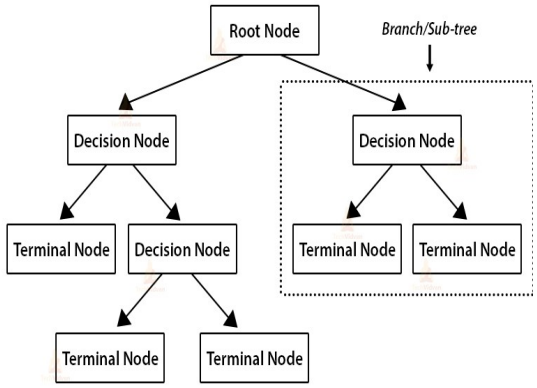

Fig. 1. Decision Tree Structure

A decision tree consists of main three components [3]: the root node, terminal node and decision node. A decision tree algorithm divides a dataset into branches, which further segregate into other branches. This sequence continues until a terminal node is attained. The terminal node cannot be differentiated further. The nodes in the decision tree represent features that are used for predicting the outcome. Different metrics like entropy, information gain, gini index help in our understanding of decision trees.

Entropy is a metric for calculating uncertainty. It is used to determine how a decision tree chooses to split data. Information gain is a measure of how uncertainty in the target variable is reduced, given a set of independent variables. The

entropy of the target variable (y) and the conditional entropy of y (given x) are used to estimate the information gain. A high information gain means that a high degree of uncertainty has been removed. Gini Index is calculated by subtracting the sum of squared probabilities of each class from one. A feature with a lower Gini index is chosen for a split.

The main difference between the decision tree algorithm and the random forest algorithm is that determining the root nodes and segregating nodes is done randomly in the random forest algorithm. The random forest uses the bagging method to generate the predictions.

Bagging involves using different samples of data rather than just one sample. The decision trees produce different outputs, depending on the training data fed to the random forest algorithm. These outputs will be ranked, and the highest will be selected as the final output. Random forest adds additional randomness to the model, while the trees are growing. It searches for the best feature among a random subset of features, rather than searching for most important feature.

While doing classification in the random forest, the training data is fed to train various decision trees. A subset of features is then chosen randomly while splitting the nodes. Every decision tree in the random forest consists of decision nodes, terminal nodes, and a root node. The terminal node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this system, the output chosen by the majority of the decision trees becomes the final output of the random forest.

One of the great quality of the random forest algorithm is ”feature importance”. It is easy to measure the relative importance of each feature on the prediction. By looking at the feature importance one can decide which features can be possibly dropped because they don’t contribute enough to the prediction process. Choosing the more important features also helps in reducing over-fitting of the data.

Advantages of random forest algorithm:

- It is versatile.It is used for both regression and classification tasks, and it’s also easy to view the relative importance of the input features.
- It can handle large datasets efficiently.
- Provides Provides a higher level of accuracy in predicting outcomes over many different methods.
- It prevents overfitting by creating random subsets of the features and building smaller trees using those subsets.

Disadvantages of random forest algorithm:

- Using a larger number of trees can make the algorithm too slow and ineffective.
- Slow in creating predictions once they are trained.

In summary, this algorithm is a great choice when one needs to develop a model quickly. It provides a pretty good indicator of the importance it assigns to the features present in the dataset. Their performance is also very hard to beat and it is a very resourceful tool for making accurate predictions. Overall, random forest is a fast, simple and flexible tool, but has small amount of limitations.

III. The Problem

The ”adult_evaluation.csv” dataset is used to identify what parameters are important for a person while buying a car. The features include buying cost, maintenance cost, no. of doors, no. of persons, size of luggage boot and level of safety. The dataset consists of 1728 observations and 7 columns.

Not much data pre-processing was required.The ”doors” column had values 2, 3, 4 and 5more, this ”5more” is replaced with ”more”. There were no missing values in the dataset. All the columns except the ”Target” variable column had equal distribution of observations.

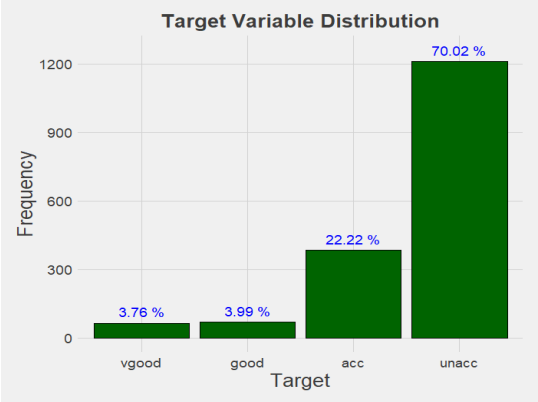

Fig. 2. Target variable Distribution

Figure 2 is the distribution of the target variable present in the dataset. It shows that around 70% of the target variable value is ”unacc”, which implies that this dataset is imbalanced.

*Data Visualization:*certain insights help in determining what parameters are most important while purchasing a car.

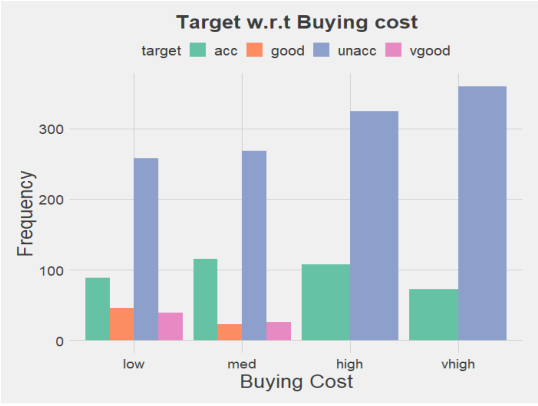

Fig. 3. Target variable w.r.t buying cost

Figure 3, shows the Target with respect to buying cost of the car. The plot suggests that most people prefer those cars whose buying cost is in low to medium range. In general as well, people prefer those cars whose buying cost is low to medium as well.

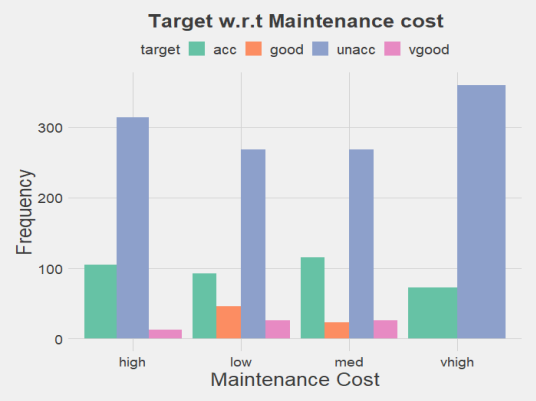

Fig. 4. Target variable w.r.t maintenance cost

Figure 4 is the distribution of the target variable with respect to maintenance cost of the car. People prefer those cars mainly whose maintenance cost is in low to medium range. People tend to prefer cars with low maintenance cost in general as well.

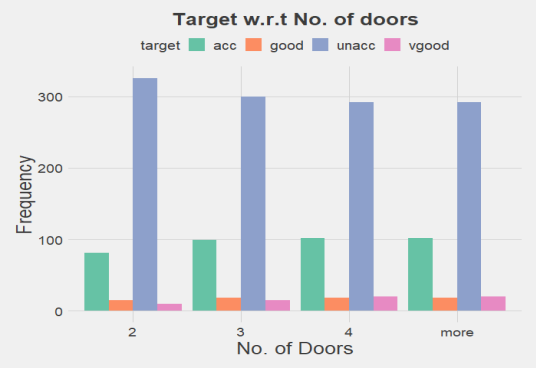

Fig. 5. Target variable w.r.t No. of doors

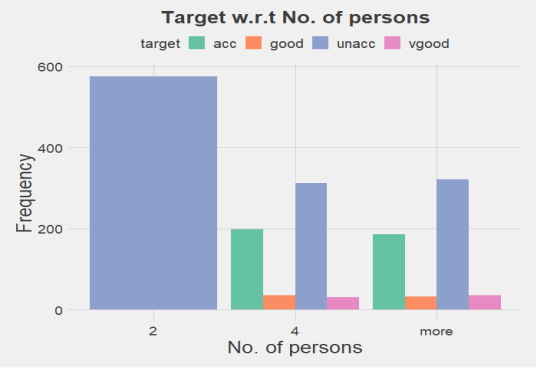

Fig. 6. Target variable w.r.t No. of persons

Figure 5 shows the distribution of target with respect to number of doors in the car. This plot shows that people are mostly indifferent to number of doors in the car, but majority of people prefer those cars which have 3 or more doors. The number of doors is not an important factor while buying a car.

Figure 6 shows the distribution of target with respect to capacity of the car. The plot shows that people prefer those cars in which 4 or more persons can sit. In accordance to the data, no one prefers cars with seating capacity of 2.

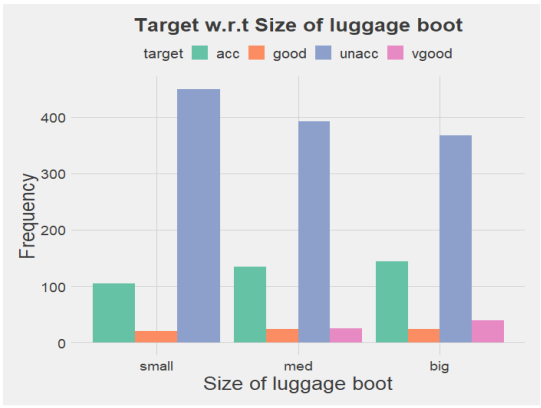

Fig. 7. Target variable w.r.t size of luggage boot

Figure 7 shows the distribution of target with respect to the size of luggage boot. People prefer cars with medium to larger luggage boot size but if the size is small it doesn't matter that much to people.

Fig. 8. Target variable w.r.t level of safety

Figure 8 shows the distribution of target with respect to level of safety provided by the car manufacturers. The plot shows that the people prefer those cars which provide higher level of safety. In general as well, people tend to prefer those cars with better equipments of safety. The level of safety variable is very likely to be an important factor while purchasing a car.

*Random Forest Model:*many different models have been built to check the different factors and to study what factors

are most important while buying a car. The dataset was split into two test data set and train data set. 1384 observations in train data set and 344 observations in test dataset. 500 trees were built in random forest.

1) Model 1: It consists of the different costs only. These costs include the buying as well maintenance cost. The Model has an accuracy of 70% but it has predicted all the 344 observations as "unacc". This model doesn't work.

```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc     0    0     0     0
     good    0    0     0     0
     unacc  76   13   242    13
     vgood   0    0     0     0

Overall Statistics

               Accuracy : 0.7035
```

Fig. 9. Confusion Matrix for Model-1

2) Model 2: 2nd model consists of seating capacity along with the costs. The model has an accuracy of 70%. The accuracy remains the same but the models predicts different values as well. This model also suggested that seating capacity is more important factor than the costs.

```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc     4    0     4     2
     good    0    0     0     0
     unacc  72   13   238    11
     vgood   0    0     0     0

Overall Statistics

               Accuracy : 0.7035
```

Fig. 10. Confusion Matrix for Model-2

3) Model 3: Along with all the other factors used in model 2, the model 3 also uses the number of doors factor. This model had an accuracy of 62% as well. There was not much learning form this model. The number of door factor does not seem to be important while choosing a car.

4) Model 4: This model has size of luggage boot added as well. The luggage boot also doesn't seem to be much important parameter while buying a car. The model had an accuracy of 65%.

```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc    30    1    37     2
     good    1    0     7     1
     unacc  45    9   194    10
     vgood   0    3     4     0

Overall Statistics

               Accuracy : 0.6512
```

Fig. 11. Confusion Matrix for Model-4

5) Model 5: This model has level of safety factor along with all the other features. This model had an accuracy of 98% suggesting that safety is an very important factor while choosing a car.

```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc    75    2     3     0
     good    1   11     0     0
     unacc   0    0   239     0
     vgood   0    0     0    13

Overall Statistics

               Accuracy : 0.9826
```

Fig. 12. Confusion Matrix for Model-5

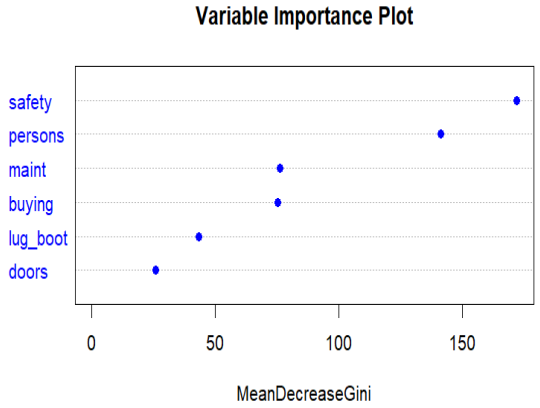

Fig. 13. Variable Importance plot

Fig 13 [4] shows the importance of variables while building up the model. According to the plot, safety and seating capacity are the most important parameters while buying the car, buying cost and maintenance cost are important but size of luggage boot and number of doors are the least important parameters while purchasing a car.

6) Model 6: The model 6 consists of buying cost, maintenance cost, safety and no of persons and luggage boot. This model had an accuracy of 95%. This much accuracy is good as well. While building up the model the number of doors variable can be ignored.

```
Confusion Matrix and Statistics

          Reference
Prediction acc good unacc vgood
     acc    73    1     7     1
     good    2   10     0     0
     unacc   1    0   235     0
     vgood   0    2     0    12

Overall Statistics

               Accuracy : 0.9593
```

Fig. 14. Confusion Matrix for Model-6

All these models indicate that number of persons and safety are the most important factor to keep in mind while buying the car, the costs should be thought after that. Number of doors in the car is not important while purchasing a car.

IV. Conclusion

Random Forest classifier helped in finding those parameters which were more important to keep in mind while purchasing a car. Seating capacity of the car, level of safety in the car are the most important factor, buying cost, maintenance costs and size of luggage boots are pretty important and number of doors doesn't matter that much while purchasing a car. The model(Model 5) without number of door factor gave an accuracy of 98% on our validation dataset. Random Forest gave a much better accuracy than decision tree model which was used previously on this dataset.

Further improvements are very likely possible. The relationship between the features was not studied that much, the relationship can help in finding better insights. Instead of only using random forest classifier, support vector machine and other sophisticated models like neural networks can be used as well.

References

[1] Awwalu, Jamilu & Ghazvini, Anahita & Abu Bakar, Azuraliza. (2014). Performance Comparison of Data Mining Algorithms: A Case Study on Car Evaluation Dataset. International Journal of Computer Trends and Technology. 13. 78-82. 10.14445/22312803/IJCTT-V13P117.

[2] https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/tutorial-random-forest-parameter-tuning-r/tutorial/

[3] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013.

[4] https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

Code Documentation

03/11/2021

Loading required packages

```
library(tidyverse)
library(randomForest)
library(caret)
library(ggthemes)
```

Reading and pre-process the data

```
data<-read_csv("car_evaluation.csv",col_names = FALSE)

colnames(data)<-c("buying", "maint","doors", "persons", "lug_boot","safety", "target")

data$doors<-gsub(pattern = "5more",replacement = "more",x=data$doors)

data<-data%>%mutate_if(is.character,as.factor)
```

Visualizations

Setting up the plotting theme

```
my_theme<-theme_fivethirtyeight()+theme(plot.title = element_text(hjust = 0.5,size=20),
                                        axis.title = element_text(size=20),
                                        axis.text = element_text(size=14),
                                        plot.subtitle = element_text(hjust=0.5),
                                        legend.position = "top",legend.title = element_text(size=15),
                                        legend.text = element_text(size=15))
```

Fig 2

```
target_var<-as.data.frame(table(data$target))

colnames(target_var)<-c("Target","Frequency")
```

```
target_var%>%mutate(Target=reorder(Target,Frequency))%>%
  ggplot()+geom_col(aes(Target,Frequency),fill="darkgreen",col="black")+
  labs(title = "Target Variable Distribution")+
  my_theme+
  geom_text(aes(x=Target,y=Frequency+50,
                label=paste(round((Frequency*100)/sum(Frequency),2),"%")),size=5,col="blue")+
  scale_y_continuous(breaks = seq(0,1200,300))
```

Fig 3

```
data$buying<-factor(data$buying,levels = c("low","med","high","vhigh"))

data%>%ggplot()+geom_bar(aes(buying,fill=target),position = "dodge")+
  labs(x="Buying Cost",y="Frequency",title = "Target w.r.t Buying cost")+
  my_theme+
  scale_fill_brewer(palette = "Set2")
```

Fig 4

```
data%>%ggplot()+geom_bar(aes(maint,fill=target),position = "dodge")+
  labs(x="Maintenance Cost",y="Frequency",title = "Target w.r.t Maintenance cost")+
  my_theme+
  scale_fill_brewer(palette = "Set2")
```

Fig 5

```
data%>%ggplot()+geom_bar(aes(doors,fill=target),position = "dodge")+
  labs(x="No. of Doors",y="Frequency",title = "Target w.r.t No. of doors")+
  my_theme+
  scale_fill_brewer(palette = "Set2")
```

Fig 6

```
data%>%ggplot()+geom_bar(aes(persons,fill=target),position = "dodge")+
  labs(x="No. of persons",y="Frequency",title = "Target w.r.t No. of persons")+
  my_theme+
  scale_fill_brewer(palette = "Set2")
```

Fig 7

```
data$lug_boot<-factor(data$lug_boot,levels =c("small","med","big") )

data%>%ggplot()+geom_bar(aes(lug_boot,fill=target),position = "dodge")+
  labs(x="Size of luggage boot",y="Frequency",title = "Target w.r.t Size of luggage boot")+
  my_theme+
  scale_fill_brewer(palette = "Set2")
```

Fig 8

```
data$safety<-factor(data$safety,levels = c("low","med","high"))

data%>%ggplot()+geom_bar(aes(safety,fill=target),position = "dodge")+
  labs(x="Level of Safety",y="Frequency",title = "Target w.r.t level of Safety")+
  my_theme+
  scale_fill_brewer(palette = "Set2")
```

Model Building

Splitting the dataset

```
set.seed(123)
Index <- createDataPartition(data$target,p=0.8,list=FALSE)

train_data<-data[Index,]
test_data<-data[-(Index),]
```

Models

Model 1

```
model1<-randomForest(target~buying+maint,data=train_data)

print(model1)

print(importance(model1,type = 2))

predict_model1<-predict(model1,test_data,type='class')


cm_model1<-confusionMatrix(predict_model1,test_data$target)
```

Model 2

```
model2<-randomForest(target~buying+maint+persons,data=train_data)

print(model2)

print(importance(model2,type = 2))

predict_model2<-predict(model2,test_data,type='class')

cm_model2<-confusionMatrix(predict_model2,test_data$target)
```

Model 3

```
model3<-randomForest(target~buying+maint+persons+doors,data=train_data)

print(model3)

print(importance(model3,type = 2))

predict_model3<-predict(model3,test_data,type='class')

test_data$preicted_target_model3<-predict_model3

cm_model3<-confusionMatrix(predict_model3,test_data$target)
```

Model 4

```
model4<-randomForest(target~buying+maint+persons+doors+lug_boot,data=train_data)

print(model4)

print(importance(model4,type = 2))

predict_model4<-predict(model4,test_data,type='class')

cm_model4<-confusionMatrix(predict_model4,test_data$target)
```

Model 5

```
model5<-randomForest(target~buying+maint+persons+doors+lug_boot+safety,data=train_data)

print(model5)
```

```
print(importance(model5,type = 2))

predict_model5<-predict(model5,test_data,type='class')

cm_model5<-confusionMatrix(predict_model5,test_data$target)
```

Variable Importance Plot

```
varImpPlot(model5, main="Variable Importance Plot",pch=16,col="blue",cex=1.3)
```

Model 6

```
model6<-randomForest(target~persons+safety+buying+maint+lug_boot,data=train_data)

print(model6)

predict_model6<-predict(model6,test_data,type='class')

test_data$preicted_target_model6<-predict_model6

cm_model6<-confusionMatrix(predict_model6,test_data$target)
```