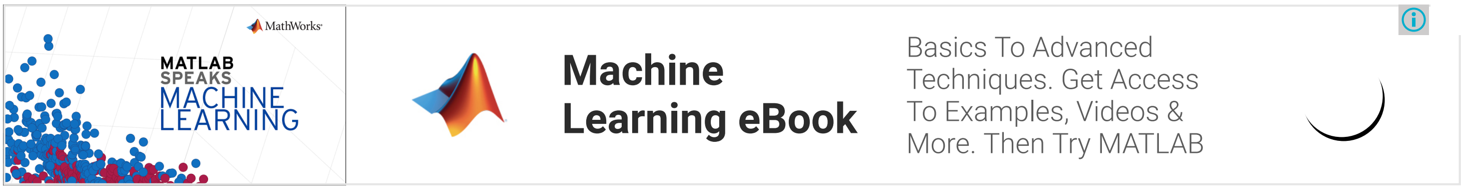
Comparing a Random Forest to a CART model (Part 2)

BUSINESS ANALYTICS (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/BUSINESS-ANALYTICS/) R (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/R/)

(http://www.facebook.com/sharer.php?u=https://www.analyticsvidhya.com/blog/2014/06/comparing-random-forest-simple-cart-
t=Comparing%20a%20Random%20Forest%20to%20a%20CART%20model%20(Part%202)) (https://twitter.com/home?
omparing%20a%20Random%20Forest%20to%20a%20CART%20model%20(Part%202)+https://www.analyticsvidhya.com/blog/2014/06/comparing-
forest-simple-cart-model/) (https://plus.google.com/share?url=https://www.analyticsvidhya.com/blog/2014/06/comparing-random-forest-
art-model/) (http://pinterest.com/pin/create/button/?url=https://www.analyticsvidhya.com/blog/2014/06/comparing-random-forest-simple-
del/&media=&description=Comparing%20a%20Random%20Forest%20to%20a%20CART%20model%20(Part%202))



Random forest is one of the most commonly used algorithm in Kaggle competitions. Along with a good predictive power, Random forest model are pretty simple to build. We have previously explained the algorithm of a random forest (Introduction to Random Forest (https://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/)). This article is the second part of the series on comparison of a random forest with a CART model. In the first article, we took an example of an inbuilt R-dataset to predict the classification of an specie. In this article we will build a random forest model on the same dataset to compare the performance with previously built CART model. I did this experiment a week back and found the results very insightful. I recommend the reader to read the first part of this article (Last article (https://www.analyticsvidhya.com/blog/2014/06/comparing-cart-random-forest-1/)) before reading this one.

Background on Dataset "Iris"

Data set "iris" gives the measurements in centimeters of the variables : sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of Iris. The dataset has 150 cases (rows) and 5 variables (columns) named Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, Species. We intend to predict the Specie based on the 4 flower characteristic variables.

We will first load the dataset into R and then look at some of the key statistics. You can use the following codes to do so.

```
data(iris)

# look at the dataset

summary(iris)

# visually look at the dataset

qplot(Petal.Length,Petal.Width,colour=Species,data=iris)
```

Results using CART Model

The first step we follow in any modeling exercise is to split the data into training and validation. You can use the following code for the split. (We will use the same split for random forest as well)

```
train.flag <- createDataPartition(y=iris$Species,p=0.5,list=FALSE)
```

```
training <- iris[train.flag,]
```

```
Validation <- iris[-train.flag,]
```

CART model gave following result in the training and validation :

Misclassification rate in training data = 3/75

Misclassification rate in validation data = 4/75

As you can see, CART model gave decent result in terms of accuracy and stability. We will now model the random forest algorithm on the same training dataset and validate it using same validation dataset.

Building a Random forest model

We have used “caret” , “randomForest” and “randomForestSRC” package to build this model. You can use the following code to generate a random forest model on the training dataset.

```
> library(randomForest)
```

```
> library(randomForestSRC)
```

```
> library(caret)
```

```
> modfit <- train(Species~ .,method="rf",data=training)
> pred <- predict(modfit,training)
```

```
> table(pred,training$Species)

  pred        setosa versicolor virginica
 setosa         25        0          0
 versicolor      0       25          0
 virginica       0        0         25
```

Misclassification rate in training data = 0/75 ***[This is simply awesome!]***

Validating the model

Having built such an accurate model, we will like to make sure that we are not over fitting the model on the training data. This is done by validating the same model on an independent data set. We use the following code to do the same :

```
> train.cart<-predict(modfit,newdata=training)
```

```
> table(train.cart,training$Species)
```

```
> train.cart   setosa versicolor virginica
```

```
  pred        setosa versicolor virginica
 setosa         25        0          0
 versicolor      0       22          0
 virginica       0        3         25
```

```
# Misclassification rate = 3/75
```

Only 3 misclassified observations out of 75, signifies good predictive power. However, we see a significant drop in predictive power of this model when we compare it to training misclassification.

Comparison between the two models

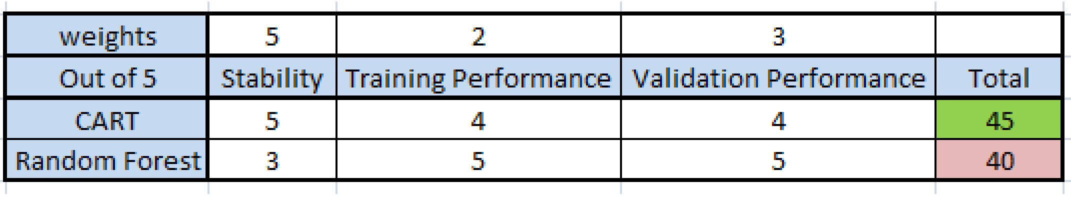
Till this point, everything was as per books. Here comes the tricky part. Once you have all performance metrics, you need to select the best model as per your business requirement. We will make this judgement based on 3 criterion in this case apart from business requirements:

.**1. Stability** : The model should have similar performance metrics across both training and validation. This is very essential because business can live with a lower accuracy but not with a lower stability. We will give the highest weight to stability. For this case let's take it as 5.

2. Performance on Training data : This is one of the important metric but nothing conclusive can be said just based on this metric. This is because an over fit model is unacceptable but will get a very high score at this parameter. Hence, we will give a low weight to this parameter (say 2).

3. Performance on Validation data : This metric catch holds of overfit model and hence is an important metric. We will score it higher than performance and lower than stability. For this case let's take it as 3.

Note that the weights and scores entirely depends on the business case. Following is a score table as per my judgement in this case.



weights	5	2	3	
Out of 5	Stability	Training Performance	Validation Performance	Total
CART	5	4	4	45
Random Forest	3	5	5	40

(https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/06/Comparison.png)As you can see from the table that however Random forest gives me a better performance, I still will go ahead and use CART model because of the stability factor. Other factor in favor of CART model is the easy business justification. Random forest is very difficult to explain to people working on field. CART models are simple cuts which can be justified by simple business justification/reasons. But the choice of model selection is entirely dependent on business requirement.

End Notes

Every model has its own strength. Random forest, as seen from this case study, has a very high accuracy on the training population, because it uses many different characteristics to make a prediction. But, because of the same reason, it sometimes over fits the model on the data. CART model on the other side is simplistic criterion cut model. This might be over simplification in some case but works pretty well in most business scenarios. However, the choice of model might be business requirement dependent, it is always good to compare performance of different model before taking this call.

Did you find the article useful? Did this article solve any of your existing dilemmas? Have you compared the two models in any of your projects? If you did, share with us your thoughts on the topic.

If you like what you just read & want to continue your analytics learning, subscribe to our emails (http://feedburner.google.com/fb/a/mailverify?uri=analyticsvidhya), follow us on twitter (http://twitter.com/analyticsvidhya) or like our facebook page (http://facebook.com/analyticsvidhya).

Share this:

(https://www.analyticsvidhya.com/blog/2014/06/comparing-random-forest-simple-cart-model/?share=linkedin&nb=1) 58

(https://www.analyticsvidhya.com/blog/2014/06/comparing-random-forest-simple-cart-model/?share=facebook&nb=1)

G+ (https://www.analyticsvidhya.com/blog/2014/06/comparing-random-forest-simple-cart-model/?share=google-plus-1&nb=1)

(https://www.analyticsvidhya.com/blog/2014/06/comparing-random-forest-simple-cart-model/?share=twitter&nb=1)

(https://www.analyticsvidhya.com/blog/2014/06/comparing-random-forest-simple-cart-model/?share=pocket&nb=1)

(https://www.analyticsvidhya.com/blog/2014/06/comparing-random-forest-simple-cart-model/?share=reddit&nb=1)