Rishabh Taneja

Northeastern University

ALY 6020: Predictive Analytics

FALL ‘18

Instructor Name: Justin Rodgers

**Part B: Random Forests**

**INTRODUCTION**:

This is an illustration of Random Forest model on credits data for identifying risky bank loans. Random forest combines the decision trees and train them on different set of observations. By using this we will compare this with the C5.0 algorithm and determine which one is the best among them depending on the accuracy.

**METHOD**:

We will make use of Random forest classifier which splits the nodes into separate trees considering limited number of attributes. The final predicted value is the average of all the predicted values of individual trees.

* 1. **Collecting data**

The credit dataset includes 1,000 examples on loans, plus a set of numeric and nominal features indicating the characteristics of the loan and the loan applicant. A class variable indicates whether the loan went into default.

* 1. **Exploring the data**

credit <- read.csv(file.choose(), header = TRUE)

str(credit)

A close up of a keyboard

Description automatically generated

From the above results, we can see that there are 1000 observations with 17 variables out of which 10 are factors and 7 and integers.

table(credit$checking\_balance)

A screenshot of a cell phone

Description automatically generated

table(credit$savings\_balance)



summary(credit$months\_loan\_duration)

A close up of a screen

Description automatically generated

summary(credit$amount)

A close up of a screen

Description automatically generated

The loan amount ranged from 250 to 18424 DM across 4 to 72 months.

summary(credit$default)



We are using this algorithm to focus on number of defaults and analyze who are at high risk and thus prevent from giving them the credit requests.

**Data preparation**

Seed set at 123 for consistent output.

train\_sample <- sample(1000, 900)

We are using the 90:10 ratio for train and test respectively.

str(train\_sample)

credit\_train <- credit[train\_sample, ]

credit\_test <- credit[-train\_sample, ]

prop.table(table(credit\_train$default))

A close up of a logo

Description automatically generated

prop.table(table(credit\_test$default))

A picture containing object

Description automatically generated

* 1. **Training the model on data**

set.seed(300)

rf <- randomForest(default ~ ., data = credit\_train)

summary(rf)

A black sign with white text

Description automatically generated

* 1. **Evaluating model performance**

credit\_pred1 <- predict(rf, credit\_test)

CrossTable(credit\_test$default, credit\_pred1,

prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,

dnn = c('actual default', 'predicted default'))

A screenshot of a cell phone

Description automatically generated

The classifier predicted 62 correct values for not default and 14 for default. The total accuracy turns out to be 62+14/100 = 76/100 = 0.76 or 76% which is quite good overall. However, when we consider the default values correctly predicted, it is only 14/33 = 42.4% which is quite bad and can be costly. In total, the classifier predicted 19+5/100 = 24/100 = 24% of the total values incorrectly.

* 1. **Improving model performance**

ctrl <- trainControl(method = "repeatedcv",

number = 10, repeats = 10)

grid\_rf <- expand.grid(.mtry = c(2, 4, 8, 16))

m\_rf <- train(default ~ ., data = credit\_train, method = "rf",

metric = "Kappa", trControl = ctrl,

tuneGrid = grid\_rf)

credit\_pred2 <- predict(m\_rf, credit\_test)

CrossTable(credit\_test$default, credit\_pred2,

prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,

dnn = c('actual default', 'predicted default'))

A screenshot of a cell phone

Description automatically generated

After the improvisation, the results are not very different. The total accuracy turns out to be 62+15/100 = 77% which is just 1% higher than the previous. However, there is no significant improvement in the prediction of default values, 15/33 = 45.4%.

1. Random Forest vs C5.0 Decision trees

**CONCLUSION and SUMMARY**: We are using this algorithm to focus on number of defaults and analyze who are at high risk and thus prevent from giving them the credit requests. Before the improvement, the classifier predicted 62 correct values for not default and 14 for default. The total accuracy turns out to be 62+14/100 = 76/100 = 0.76 or 76% which is quite good overall. However, when we consider the default values correctly predicted, it is only 14/33 = 42.4% which is quite bad and can be costly. In total, the classifier predicted 19+5/100 = 24/100 = 24% of the total values incorrectly. After the improvement, there were no significant improvement just that the accuracy improved by 1%.

**DISCUSSION**:

As we can note from Part A, before the improvisation of the model, the classifier had predicted 73 correct values, 14 for default and 59 for not default with 73% accuracy. Whereas, in Part B, the classifier predicted 76 correct values, 14 for default and 62 for not default with 76% accuracy. Overall, these two classifiers did not have much difference in the correct prediction, however, when we consider correct prediction of values which were default in actual and predicted as yes, both the model have the same accuracy for it which is 14/33 = 0.42 or 42%.

Considering after the improvisation, the step made the C5.0 classifier worse as the total accuracy came down to 63%. The only improvement is in the correctly predicted defaults which is 26/33 = 78.7%. On the other hand, the random forest classifier had no significant change, but the accuracy improved by 1% from 76% to 77%.

Overall, if we were to compare the accuracy of both the classifiers just based on accuracy, the random forest classifier would win as it has a better prediction accuracy. But it would be biased to simply state this as the accuracy depends on a lot of factors as well where in C5.0 we simply consider the predicted value and in the random forest, we consider the average of the predicted values.

**REFERENCES:**

* 1. Lantz, B. (n.d.). Machine Learning with R - Second Edition. Retrieved from <https://www.oreilly.com/library/view/machine-learning-with/9781784393908/>
  2. (n.d.). Retrieved from <https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.modeler.help/c50node_general.htm>
  3. Donges, N., & Donges, N. (2018, February 22). The Random Forest Algorithm. Retrieved from <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>