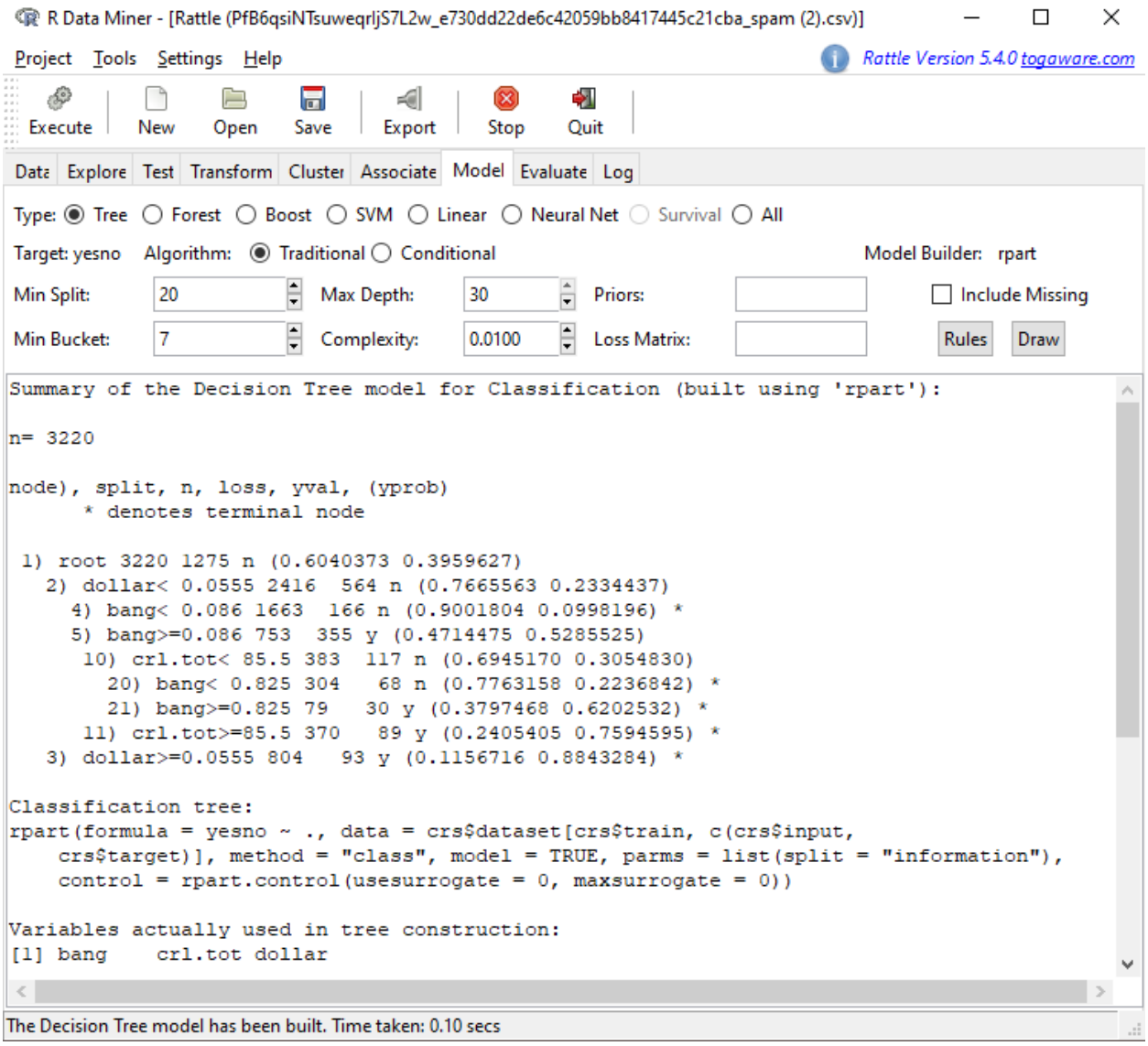
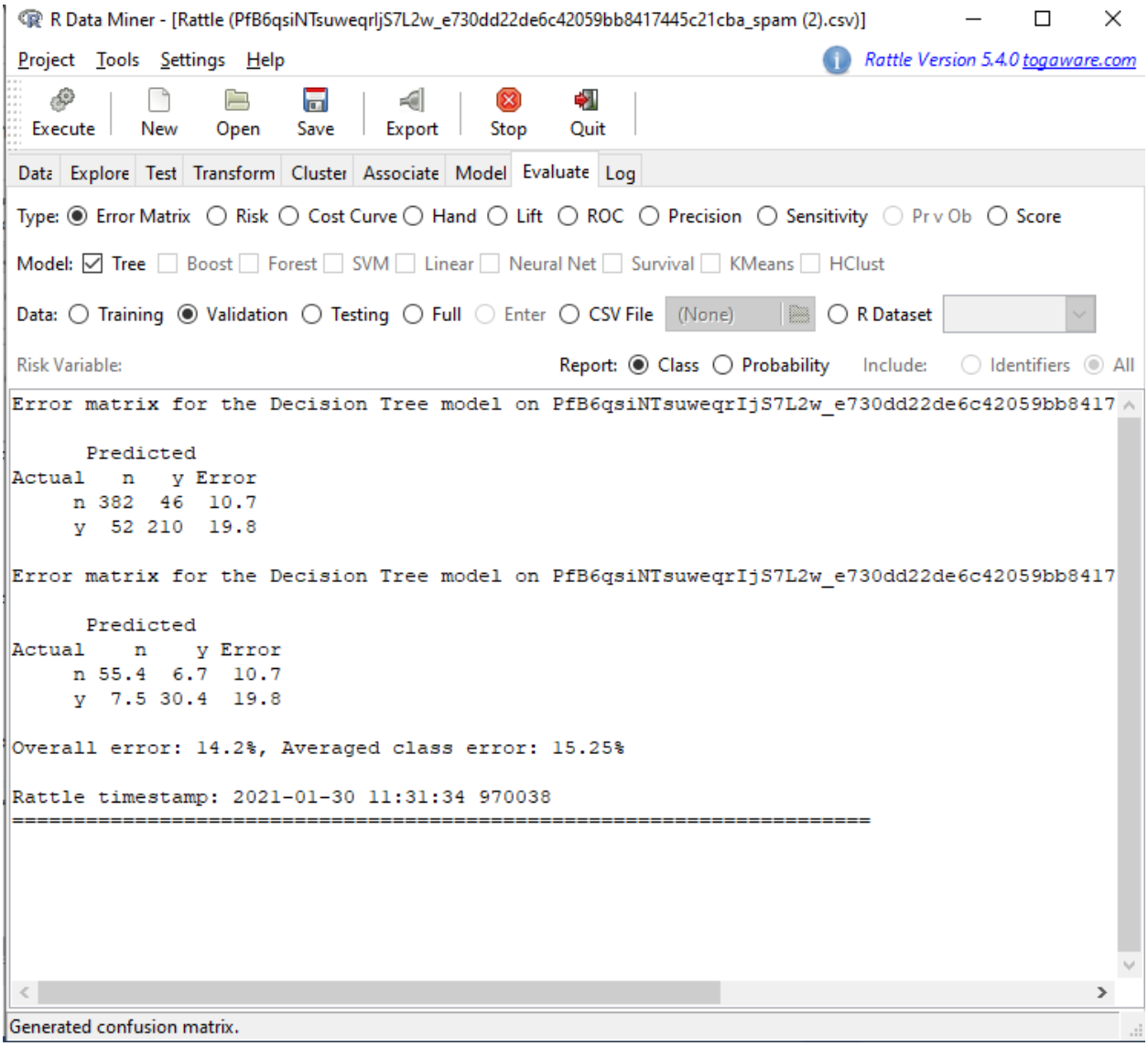
Predictive Analytics and Data Mining
Peer-graded Assignment: Module 2 Peer Review Assignment
Aya Anisa Dwinidasari – January 30th, 2021

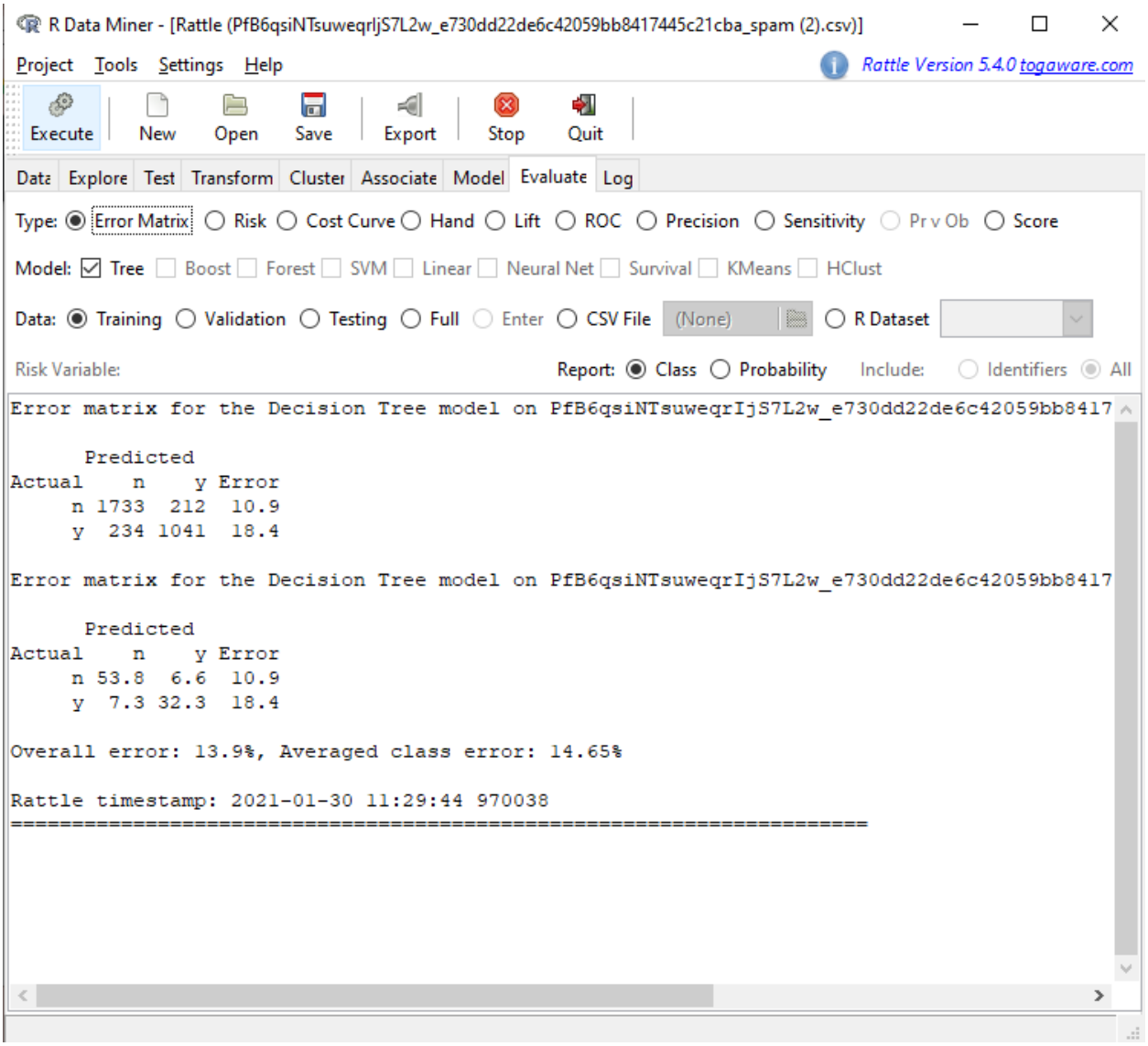
1. First, I input the data into the rattle with default partition and "yesno" as the target as shown on figure below.



2. Evaluate Step.
   After input the data, I evaluate tha error by using VALIDATION as shown on the figure below.



After I did a validation data, the next step is comparing the error result with the TRAINING mode as shown on the figure below. The VALIDATION and TRAINING mode would give different error result.



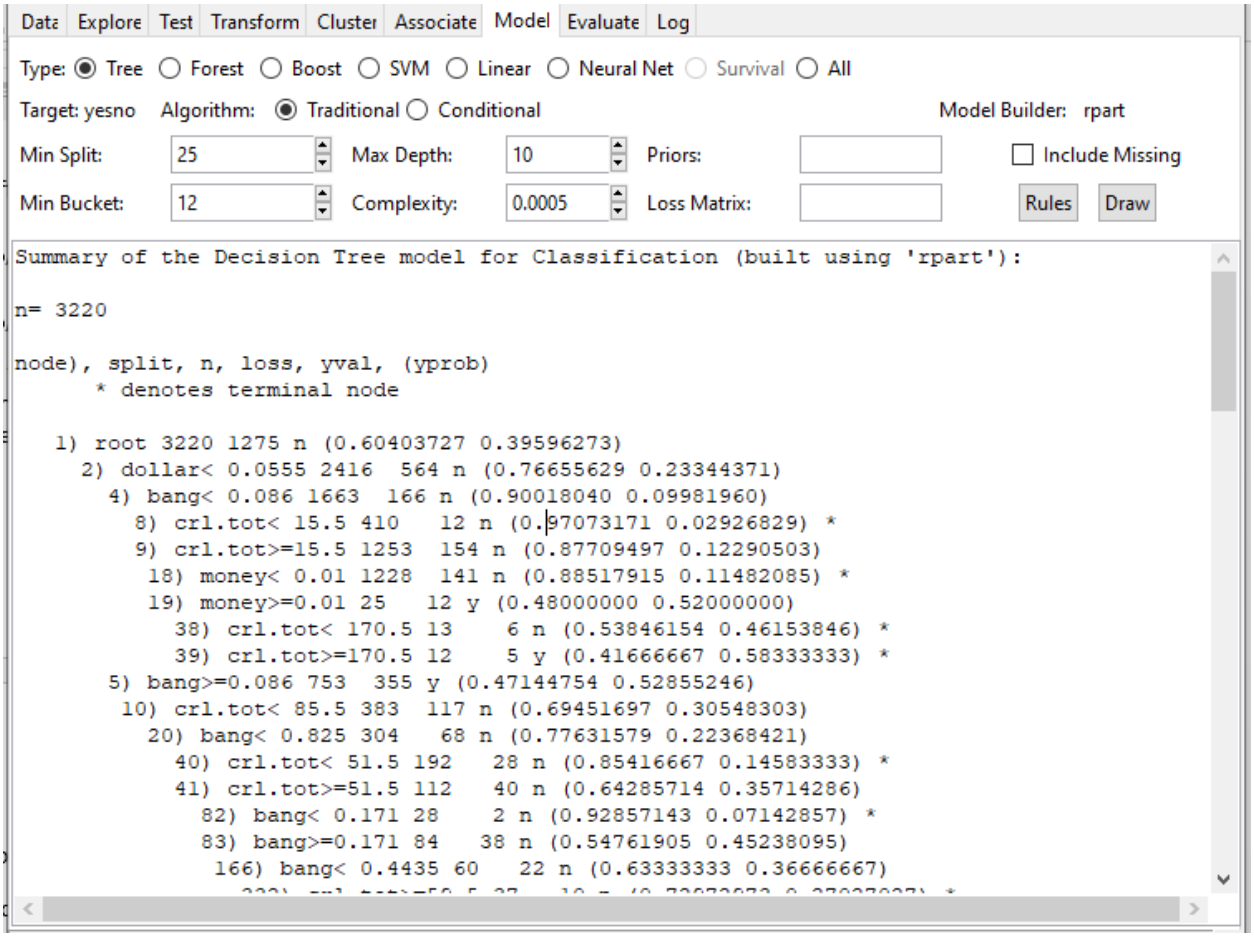
3. Error Comparison
   In this section we get two different errors from those two methods (Training and Validation)
   - **VALIDATION:**
     Overall error: 14.2 %, Averaged class error: 15.25%
   - **TRAINING:**
     Overall error: 13.9 %, Averaged class error: 14.65%

As the matter of fact, we define that the **error rate** from **VALIDATION** is **higher** than the TRAINING. The higher the error rate indicates that there are overfitting or too complex model. Overfitting is caused by many variance and noise in the model, so we should reduce the overfitting to get less error rate.

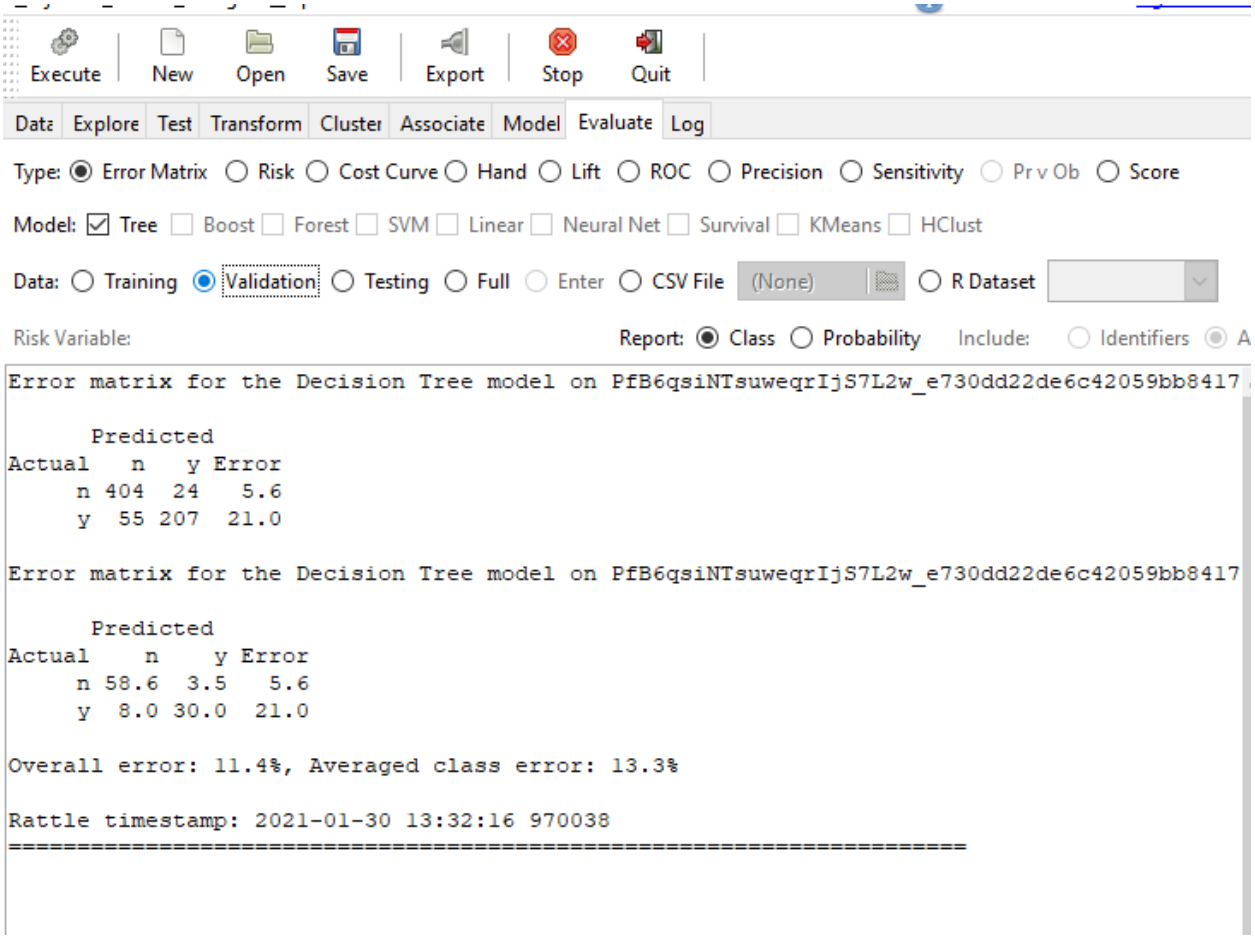
There are some ways to reduce the overfitting:

- Increase MIN SPLIT, to control the node to be less split.
- Increase MIN BUCKET, to control the tree from expanding too bigger.
- Increase COMPLEXITY PARAMETER (CP), to control the parameter of tree too be less growing.
- Decrease MAX DEPTH, to control the tree not too be too depth/big.
- Lets try changing the CP number.

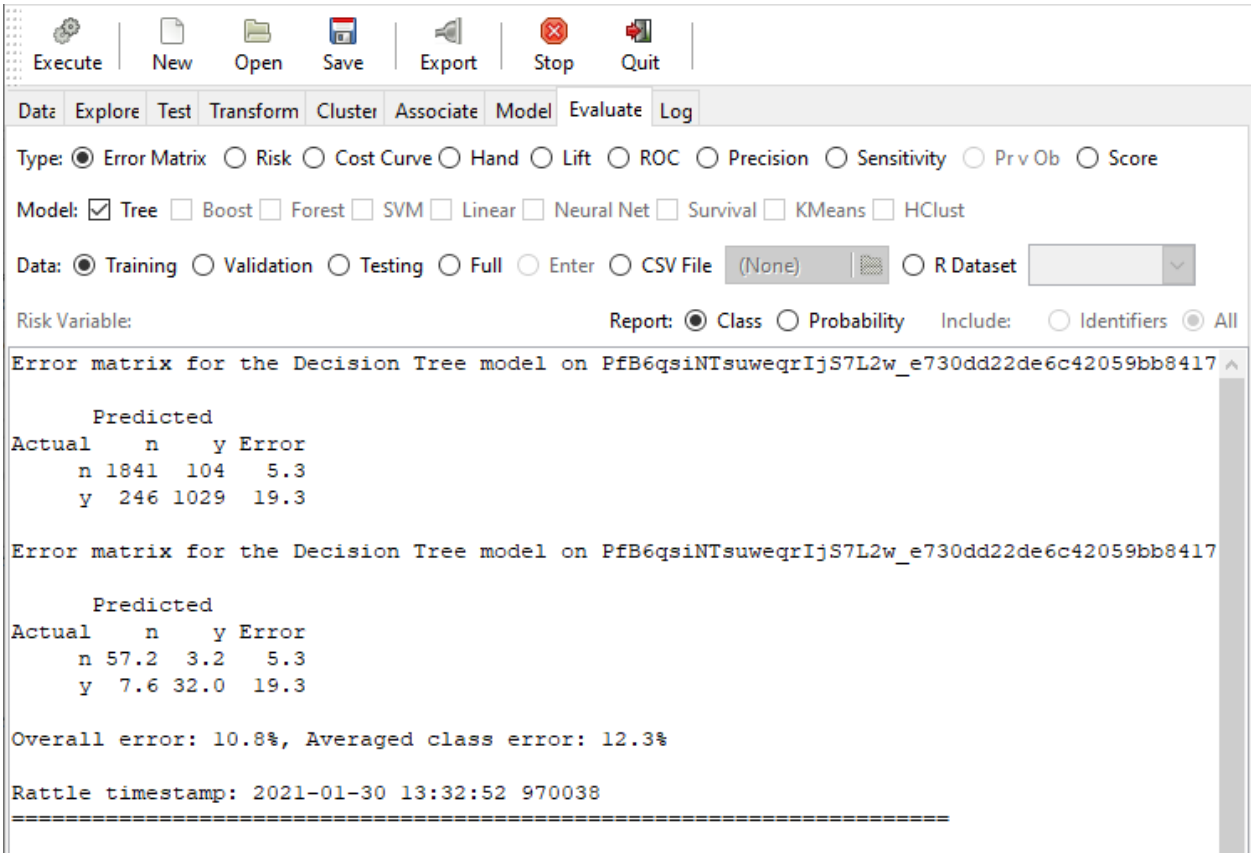
So I did some iterartion in those four parameters to reduce the overfitting, with the model parameter as shown on the figure below.



Then I did Validation evaluation from the new model.



After that, I did TRAINING Evaluation from the model too.



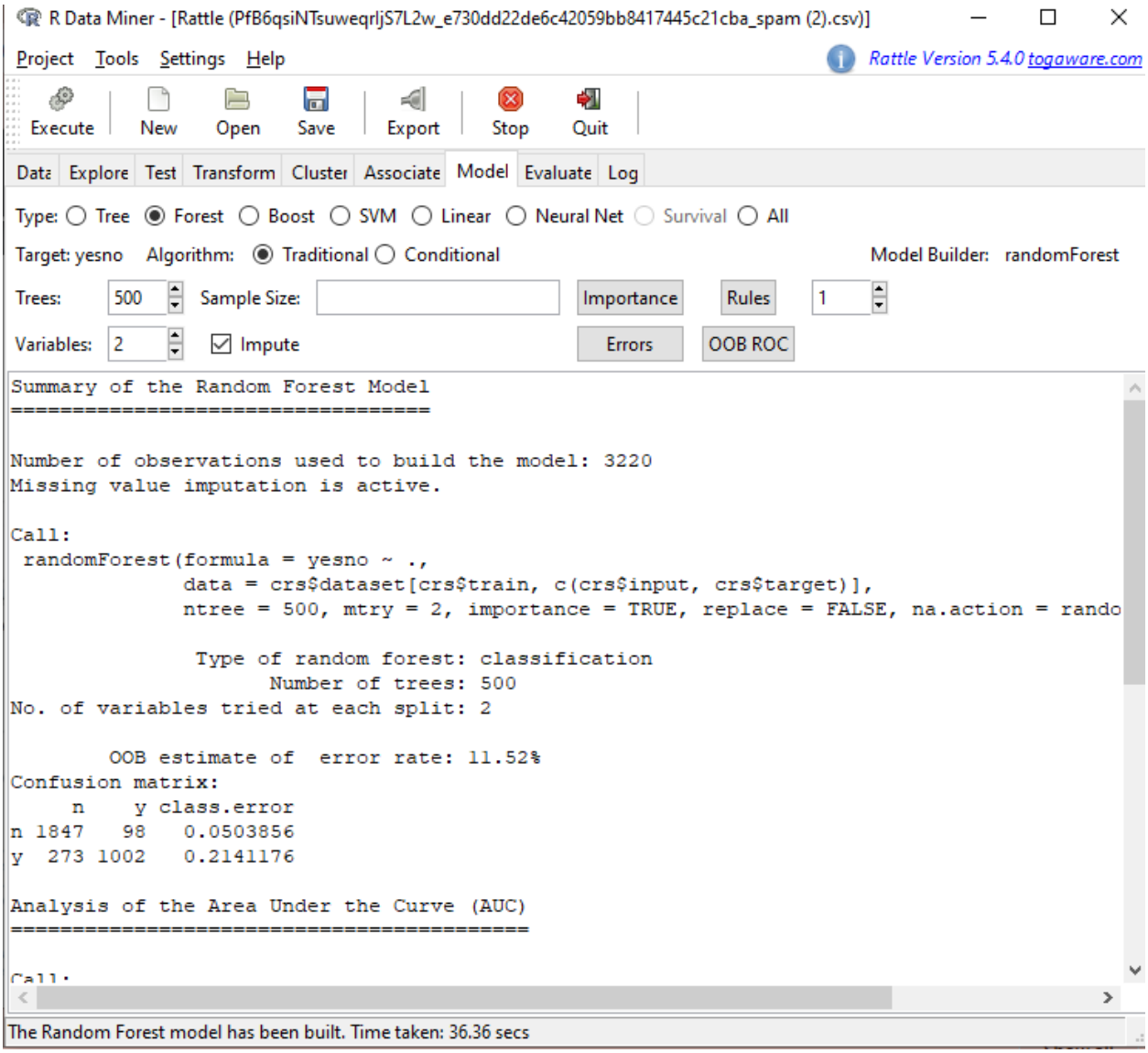
From the new model, I got new error rate, there are:

- **VALIDATION:**
  Overall error: 11.4 %, Averaged class error: 13.%
- **TRAINING:**
  Overall error: 10.8 %, Averaged class error: 12.3%

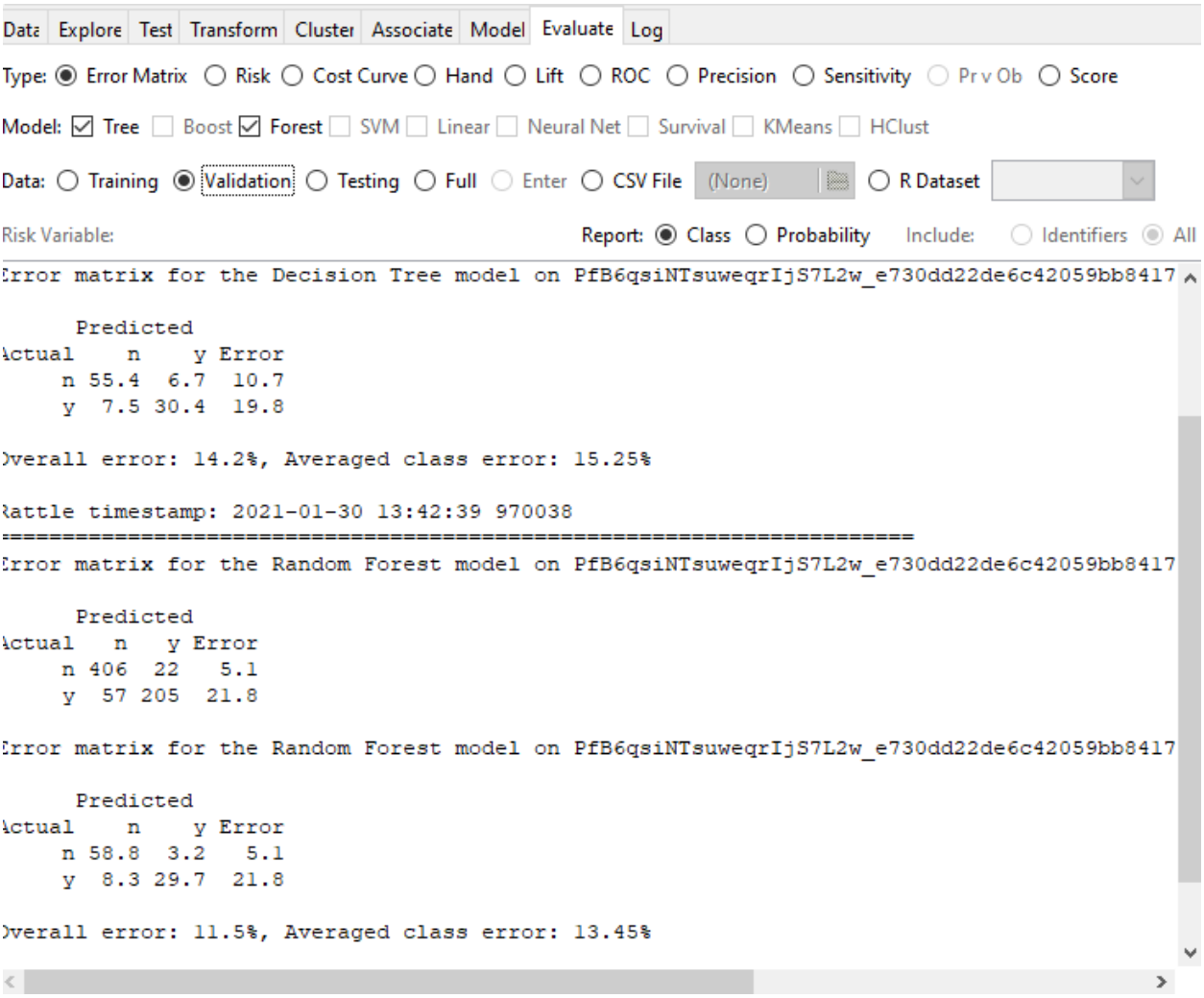
From the new model above, we clearly get new smaller error rate than the previous model.

4. Forest

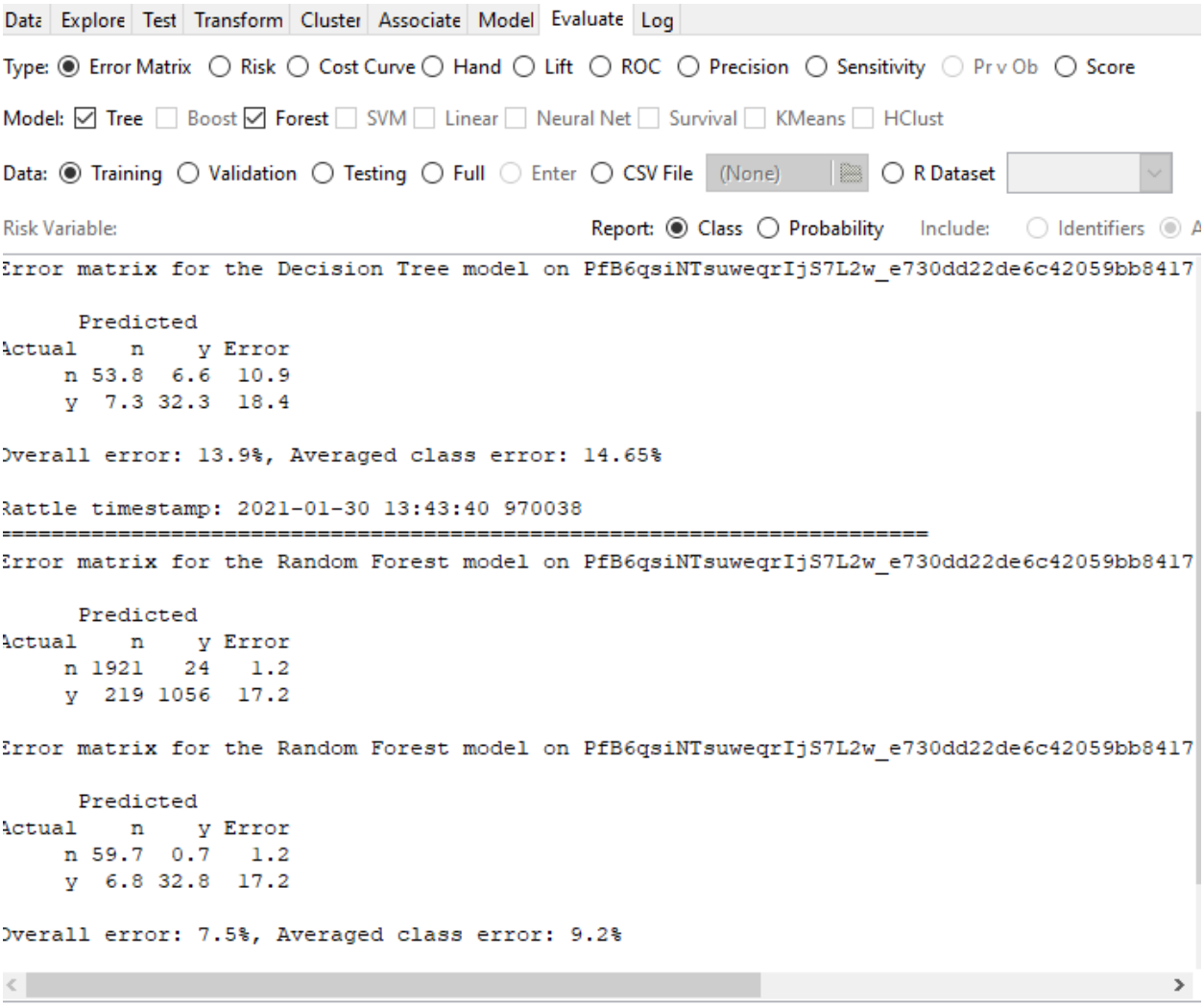
Firsr, I built the forest using defaullt setting from Rattle, as shown on the figure below.


R Data Miner - [Rattle (PfB6qsiNTsuweqrljS7L2w_e730dd22de6c42059bb8417445c21cba_spam (2).csv)]
Project Tools Settings Help
Rattle Version 5.4.0 togaware.com
Execute New Open Save Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Type: Tree Forest Boost SVM Linear Neural Net Survival All
Target: yesno Algorithm: Traditional Conditional
Model Builder: randomForest
Trees: 500 Sample Size: Importance Rules 1
Variables: 2 Impute Errors OOB ROC
Summary of the Random Forest Model
==========
Number of observations used to build the model: 3220
Missing value imputation is active.
Call:
randomForest(formula = yesno ~ .,
data = crs$dataset[crs$train, c(crs$input, crs$target)],
ntree = 500, mtry = 2, importance = TRUE, replace = FALSE, na.action = rando
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 2
OOB estimate of error rate: 11.52%
Confusion matrix:
n y class.error
n 1847 98 0.0503856
y 273 1002 0.2141176
Analysis of the Area Under the Curve (AUC)
==========
Call:
The Random Forest model has been built. Time taken: 36.36 secs

VALIDATION EVALUATION



TRAINING EVALUATION



As we see from the evalation, I did training and validate error ratte comparison. It can be seen from the figure above that the **Overall** error from the **Random Forest Model is smaller** than the Decision Tree Model. But the disadvantage of the Random Forest Model is we cannot get a clear interpretation model like shown in the decision tree model.