**Feature Engineering Scripts Guide**

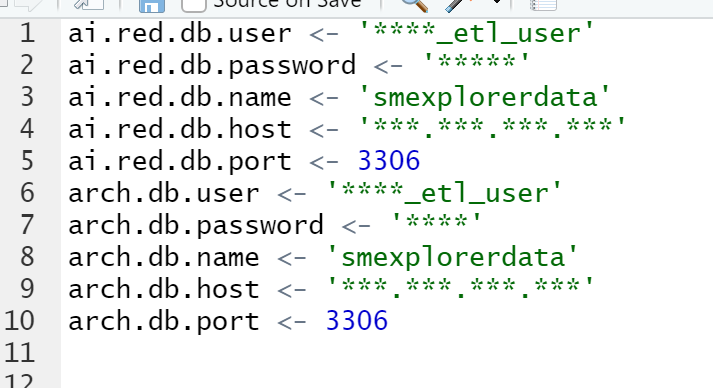
**Prerequisite**

To log into R server in devbox, you need to get RSA Token for the password to login OR get the R studio access from infra on your respective account. (Differs for every account, your respective AI member can help you with this)



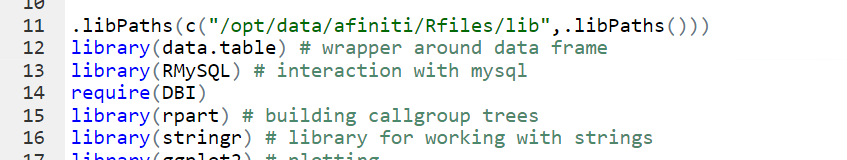
**Data preparation**

1. Change the config scripts in the folder of feature engineering to your queue’s configure (AI Server)



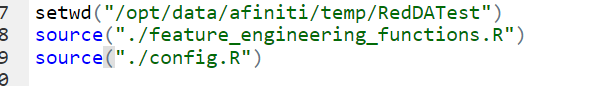
1. Put this line before library any package to get the package.

.libPaths(c("/opt/data/afiniti/Rfiles/lib",.libPaths()))

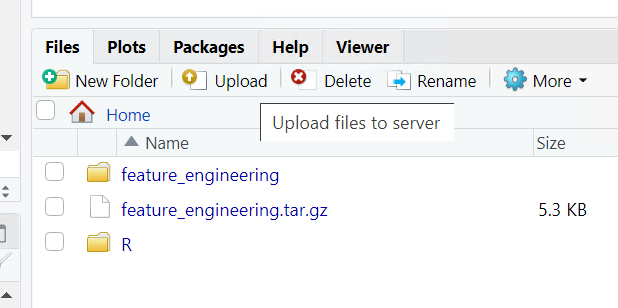


1. Set working directory to a folder in your client’s folder.

setwd("/opt/data/afiniti/temp/RedDATest")

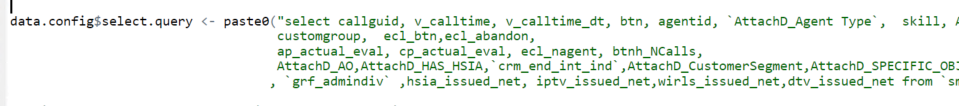


And upload the two R scripts in feature engineering’s folder to the working directory



1. Select variable that you want to test in the tree from either the sme view (Ai server) or vaca in (archival server). Variable name could differ by client and queue.   
   \*Important: Make sure you have the optimization metric in your variables as it will be required in later steps.

Line37-42

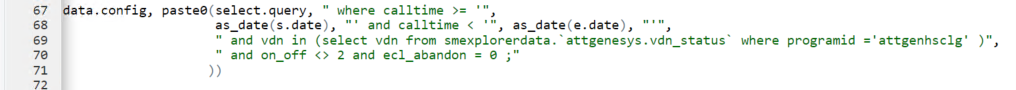


Test the scripts in MySQL workbench first to be sure it can fetch the data

1. Line52-53. Change the dates for which the required data needs to be fetched.

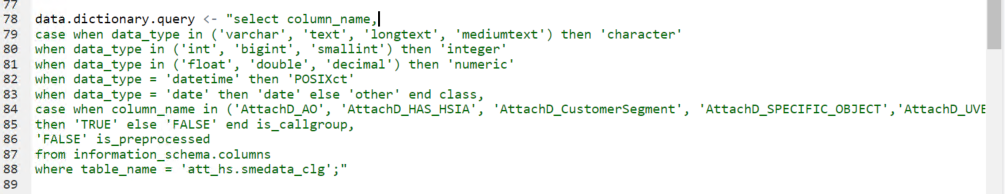


1. Line61-73. Generate query for each day.

Line67-71. 

This is optional – again, respective to each client and queue. Add in the clauses whichever required in your case. If no clauses are required then just remove line 69-70 and end the query right after the date clause.

1. Line78-88.

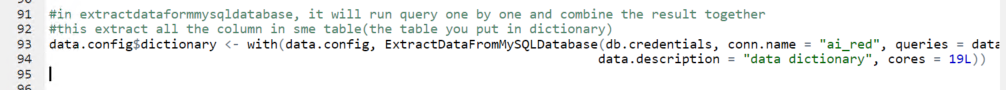


(1)Change line 84 to the variable you want to test in the tree (must be selected in previous query)

(2)Change line 88 to the table you select data from – mentioned in previous query

1. line91-94.

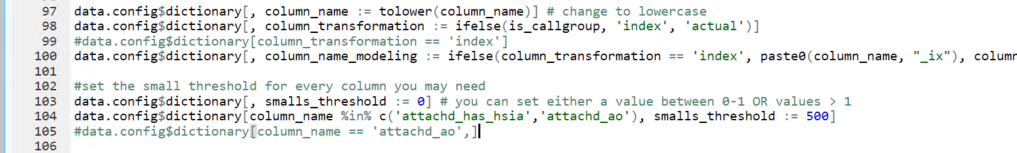
Change the cores = number of days has been selected.  
Again – this is optional depending on the size of data that is being fetched, better not to roast all the cores and leave your AI member hanging clueless on what’s happening. ;)



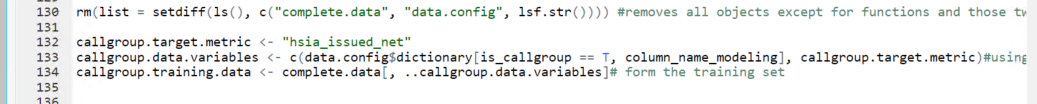
1. line 97-105.

Set Smalls threshold for the variable you put in the tree. If you do not specify, the smalls threshold will be 0.

Default setting for smalls\_threshold: 200-500. All Smalls\_threshold add up to be 5% of the total calls.



1. line 130-134. Set up outcome variable and training dataset.



Line 130. removes all objects except for functions and those two table (complete.data & data.config)

Line 132. Change the target.metric to the outcome variable in your queue

**Tree Building**

1. Line142-144.Build the tree(Post-pruning Tree)

Note:

(1)minsplit is “the minimum number of observations that must exist in a node (set it as the 1% of data in Umair’s script (N\*0.01))

(2)The complexity parameter (cp) in rpart is the minimum improvement in the model needed at each node. will stop when there are no improvment in the model

(3)anova:separate group base on variance

(4)maxdepth: The maximum depth of any node of the final tree, with the root node counted as depth 0.



Default setting values:

(1)cp = 0 (add nodes until no improvment)

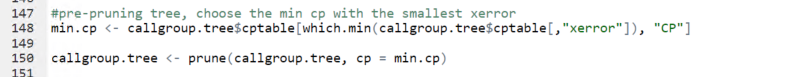
(2)maxdepth = 10~20 (could be change to test the tree). If you are using 64 bit server, set maxdepth<64; if you are using 32 bit server, maxdepth<32

(3)xval = 5(20% of your data is testing data, 80% is training data ). If you want 10% testing data, set xval = 10.

(4)maxsurrogate = 0, maxcompete = 0, method = ‘anova’

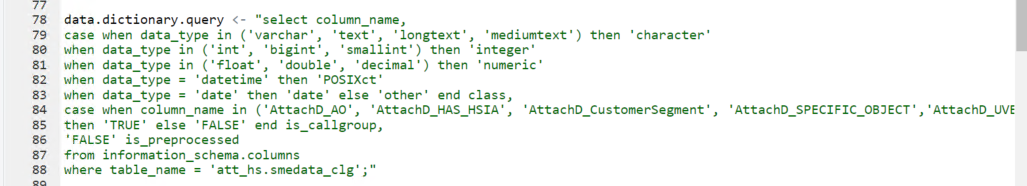
1. Line147-150.(Pre-pruning tree)

Find the best minimum cp by xerror and pruning the tree to the min.cp



1. Line 152. Find the importance of the variable you put in. 

Note: By changing the variable you put in callgroup in line84. The importance of the variable could change in different combination of variables.

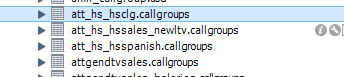


1. Line155. Add column: Give each call a call group



**Tips:**

Find the callgroup already being used by AI.



Find the callgroup table in AI server. For v5, filter that table to iscurrent = 1 to see the column being used by AI. For v6, you can get the callgroups from the json column from the callgroup table in afiniti\_3 schema.

In predicate column. Inside the CustomAttribute is the variable’s name.(e.g. AO)

