EDA VIIII Analysis
David Hunte
7/31/2018

Method

Random forest using down-sampling to encourage no data loss unlike the custom down-sampling methods that throws away information within the majority class.

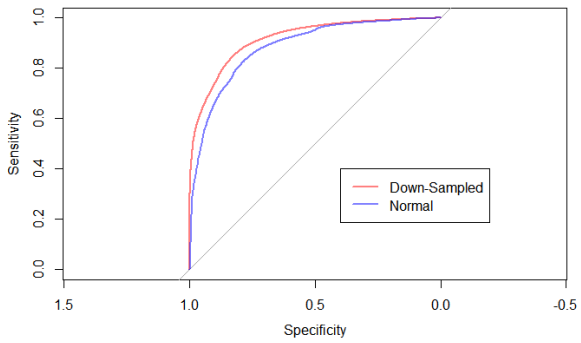
A larger part of the bootstrap samples is taken from the training data and separate unpruned trees is created for each data set and the area under the ROC curve will quantify the model effectiveness.

Consider this for the ROC curve

.90-1 = excellent (A)
.80-.90 = good (B)
.70-.80 = fair (C)
.60-.70 = poor (D)
.50-.60 = fail (F)

Explain your hyper-parameters in H20.randomforest and your results.

I used a sequence of 100, 1000, and 200 trees since a larger number of trees will produce more stable models and covariate importance estimates. Also, a depth of 30 max to have more robust trees with a small leaf size of 2



Improvement using the RF sampling procedure given a descent ROC curve pointing to reject the null hypothesis. It has a strong sensitivity…. The model has correctly predicted the variables involved, but the down-sampling method proved to be the best.

TrainROC <dbl>	TrainSens <dbl>	TrainSpec <dbl>	method <chr>
0.9321667	0.9756667	0.5166667	rf

Logloss numbers is looking good since it is very low or closer to zero showing classification of the RF Model is healthy in terms of a classification accuracies.

	max_depth <chr>	ntrees <chr>	model_ids <chr>	logloss <chr>
1	11	900	rf.grid_model_19	0.003519959285996883
2	11	700	rf.grid_model_15	0.003519959285996883
3	11	300	rf.grid_model_7	0.003519959285996883
4	11	500	rf.grid_model_11	0.003519959285996883
5	11	100	rf.grid_model_3	0.0035363662701978233
6	8	300	rf.grid_model_6	0.003710169187427133
7	8	500	rf.grid_model_10	0.003710169187427133
8	8	700	rf.grid_model_14	0.003710169187427133
9	8	900	rf.grid_model_18	0.003710169187427133
10	8	100	rf.grid_model_2	0.0037291343992123006

AUC

The fit looks very good for the top 10 model and may be a little bit too good. This tells us that there is a high recall and precision value concerning a low false negative and positive rate and these top 10 models predicts the classes the best

I could run a larger sample size, which would make some changes, but overall I will still receive a healthy model fit. 0.5 to 1 is the target, so the RF Model has done a good job here.

	max_depth	ntrees	model_ids	auc
	<chr>	<chr>	<chr>	<chr>
1	11	100	rf.grid_model_3	0.9783308874424436
2	11	900	rf.grid_model_19	0.9778970076672016
3	11	700	rf.grid_model_15	0.9778970076672016
4	11	300	rf.grid_model_7	0.9778970076672016
5	11	500	rf.grid_model_11	0.9778970076672016
6	8	100	rf.grid_model_2	0.9736591685370934
7	8	300	rf.grid_model_6	0.9736183812536799
8	8	500	rf.grid_model_10	0.9736183812536799
9	8	700	rf.grid_model_14	0.9736183812536799
10	8	900	rf.grid_model_18	0.9736183812536799

Best Model###

Looking at my best models (rf.grid_model_3)

Very nice log loss that is pretty much zero

```
Model Details:
==============

H2OBinomialModel: drf
Model Key:  rf.grid_model_3
Model Summary:

H2OBinomialMetrics: drf
** Reported on training data. **
** Metrics reported on Out-Of-Bag training samples **

MSE:  0.0003777929
RMSE:  0.0194369
LogLoss:  0.002457316
Mean Per-Class Error:  0.08219844
AUC:  0.9706374
Gini:  0.9412748
```

RMSE and the MSE

Measure of accuracy and its value is closer to zero to indicate a great fit

AUC

The RF Model fit looks very good and may be too good, showing a good fit since it is closer to the max conditional fit value (1)

```
Gains/Lift Table: Extract with `h2o.gainsLift(<model>,
<data>)` or `h2o.gainsLift(<model>, valid=<T/F>, xval=<T/F>)`
H2OBinomialMetrics: drf
** Reported on validation data. **

MSE:  0.0005262026
RMSE:  0.02293911
LogLoss:  0.003536366
Mean Per-Class Error:  0.105575
AUC:  0.9783309
Gini:  0.9566618
```

Logloss

Shows great performance by the RF Model since its values is very low or closer to zero proving the prediction classifications of the RF Model is healthy in terms of a classifications.

The Gini

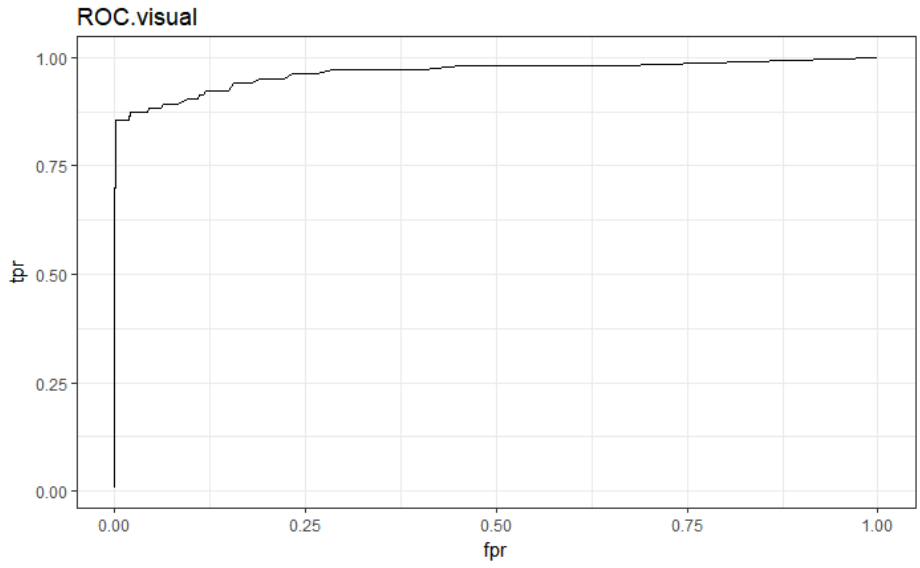
Shows some degree of high inequality within the model classifications since it is closer to 1 and not zero

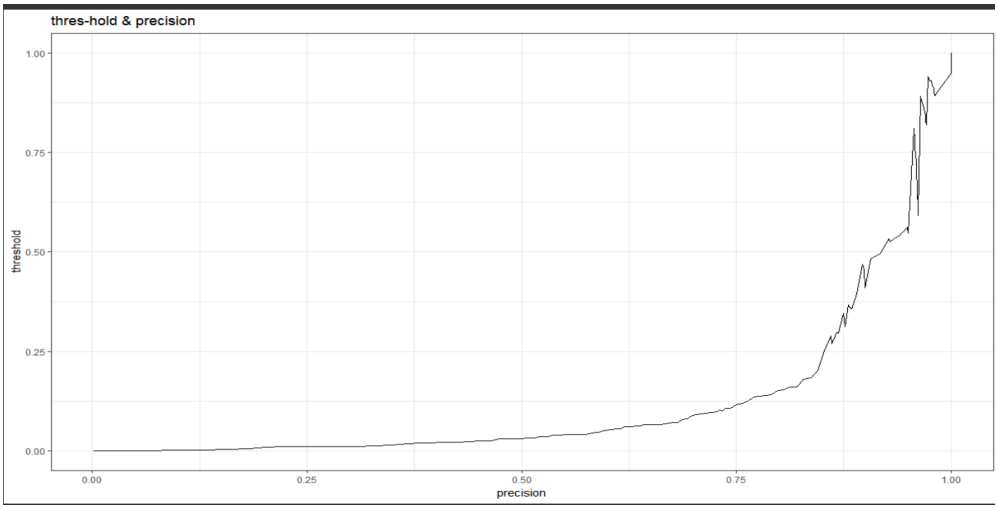
Variable of Importance

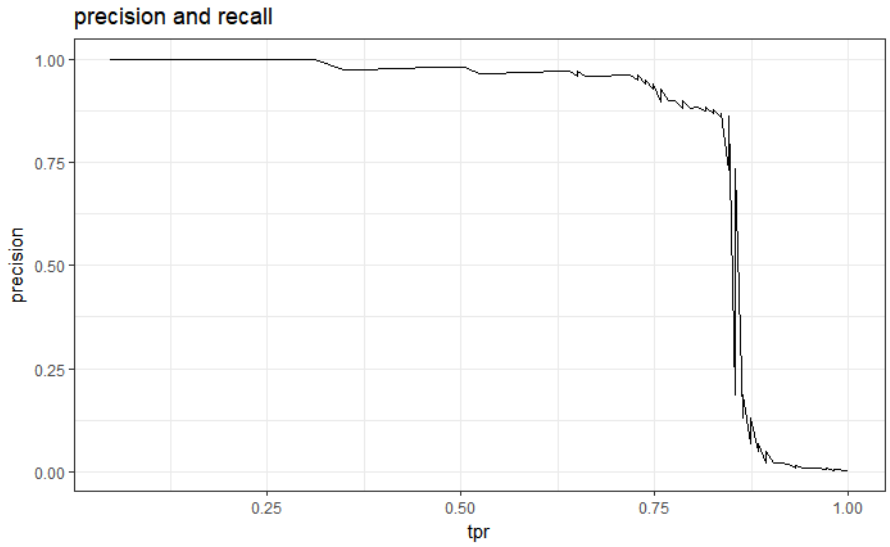
These are the most prominent variables with v17 being the best one. This shows the best variables that I can use to build my model for the best predictor.

	variable	relative_importance	scaled_importance	percentage
	<chr>	<chr>	<chr>	<chr>
1	V17	3597.381836	1.000000	0.187235
2	V12	2874.297119	0.798997	0.149600
3	V14	2502.508057	0.695647	0.130249
4	V11	1868.645508	0.519446	0.097258
5	V16	1673.236938	0.465126	0.087088

This is the ROC curve of the best model (0.97) operating characteristics concerning the true positive rate or high sensitivity rate for different cut-off points.
The threshold and precision curve follow suit to the ROC curve results showing the tradeoffs for an increasingly high precision and threshold at the tradeoff spots.







The Precision curve does not correlate with the ROC performance. Some instability exists between the low false positive rate amongst the two curves.