Should try normalised variables with and without one-hot encoding?

Good description on why to prune.

<https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/>

<https://www.analyticsvidhya.com/blog/2016/01/xgboost-algorithm-easy-steps/>

Good example of how to prune with RPART

<https://www.edureka.co/blog/implementation-of-decision-tree/>

<https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Classification/Decision_Trees>

Random Forest

Documentation of RandomForest

<https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>

A start agrees with starting with adjusting mtry then nTree

<https://www.r-bloggers.com/how-to-implement-random-forests-in-r/>

A very good walk through with detailed tuning using grids:

<https://uc-r.github.io/random_forests>

Mentions one-hot encoding

Another example of tuning RandomForest with which parameters to consider:

<https://www.hackerearth.com/blog/developers/practical-tutorial-random-forest-parameter-tuning-r/>

XGBoost

An interesting use of grid to tune XGBoost

<https://datascienceplus.com/chaid-vs-ranger-vs-xgboost-a-comparison/>

Excellent look a basic parameter tuning:

<https://insightr.wordpress.com/2018/05/17/tuning-xgboost-in-r-part-i/>

<https://www.r-bloggers.com/tuning-xgboost-in-r-part-ii/>

And this one:

<https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/beginners-tutorial-on-xgboost-parameter-tuning-r/tutorial/>

and this one uses grids:

<https://datascienceplus.com/extreme-gradient-boosting-with-r/>

Documentation by XGBoost

<https://github.com/tqchen/xgboost/blob/master/doc/how_to/param_tuning.md>

<https://github.com/tqchen/xgboost/blob/master/doc/parameter.md>

<https://datascience.stackexchange.com/questions/9364/hypertuning-xgboost-parameters>

Paper by the author of XGBoost:

<http://proceedings.mlr.press/v42/chen14.pdf>

I couldn’t access this link at GOSH

<https://rpubs.com/ippromek/336732>

A good page on SVM

<https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Classification/SVM>