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

## Tuning guides for all models

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>

## Style guides

<https://stackoverflow.com/questions/1266279/how-to-organize-large-r-programs>

<https://google.github.io/styleguide/Rguide.xml>

## Project style/layout guides

<https://career-resource-center.udacity.com/portfolio/data-science-reports>

<http://blog.kaggle.com/2016/06/29/communicating-data-science-a-guide-to-presenting-your-work/>

<https://smallbusiness.chron.com/write-data-report-61330.html>

<https://www.dataquest.io/blog/data-science-project-style-guide/>

## GGPlot Multiple lines

<https://stackoverflow.com/questions/17150183/plot-multiple-lines-in-one-graph>

<http://www.sthda.com/english/wiki/ggplot2-line-types-how-to-change-line-types-of-a-graph-in-r-software>

<https://stackoverflow.com/questions/27350243/ggplot-line-graph-with-different-line-styles-and-markers>

## SVM vs XGBoost

<https://community.alteryx.com/t5/Data-Science-Blog/Why-use-SVM/ba-p/138440>

<https://www.researchgate.net/post/When_we_use_Support_Vector_machine_for_Classification>

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

<https://www.quora.com/Why-does-XGBoost-perform-better-than-SVM>

<https://towardsdatascience.com/a-complete-view-of-decision-trees-and-svm-in-machine-learning-f9f3d19a337b>