# Module 7 Assignment 1- Project Part 2 of 2| Project Plan & Proposal

## Introduction

For this project, I am imagining a project being conducted by a somewhat recently established analytics team or a new, modern methods (machine learning) oriented, forecasting department at a hedge fund or a financial services company which provides a Roth IRA to the public. The goal for this project is for this team or department is to begin the creation of a framework they can use to choose which analytics methods/algorithms they are going to prefer using in their internal forecasting models for both risks and returns for portion of their portfolio held in stocks going forward.

The repository I chose to use as my source data for this project, which I found on Kaggle, was assembled using only the publicly available data found in the 10-K filings each publicly traded company releases annually. Therefore, there are no data privacy or security issues at play in this big data analytics project.

## Overall Big Data Analytics Project Plan

The plan for my analysis of this stock market data is to use it as a vehicle through which I can compare the performance of roughly 5 or 6 Classification Models and a similar amount of Regression Models because the dataset is capable of accommodating both in such a way that whichever 1 or 2 of the half a dozen Regression and Classification models makes the best predictions will likely do well, but this will enable me to draw some tentative lessons about the relative strengths and weaknesses of each of the models I apply to this dataset for the purpose of forecasting.

For Part I of my predictive modeling and analysis, that which takes place in the Classification setting, I will employ 5 Supervised Classification Algorithms, and 1 Unsupervised; namely:

- Logit
- PLS-DA
- Elastic Net
- Support Vector Machine
- Artificial Neural Network
- Average Neural Nets
- KNN

As of the time this report is being turned in at the end of week 7 of the course, all 7 of the chose classification modeling techniques have been run successfully in R using the Caret package.

As for Part II, forecasting the behavior of the variance in stock prices in the following year(s) based off of the predictor values trained on this year's data, I employed the following regression techniques:

- Ridge Regression
- MARS
- Artificial Neural Network
- Average Neural Nets
- SVM
- Bagged SVMs
- Random Forest
- Traditional Multiple Regression Analysis

At the time this report was completed, on Saturday, December 3rd, 2022, I had only been able to get the R code for only the Ridge, MARS, and Random Forest Regressions to run; however, that is largely

because about a week ago, I decided to focus all of my effort on the Classification models so that at least one of the two parts of the analysis would be mostly complete. So, this was largely deliberate.

The nice thing about this source data repository is that because there are 5 separate equivalent datasets, it will be easier to perform several different layers of cross-validation, and model-retraining without even having to use sample splitting, K-Fold Cross-Validation, or re-sampling methods to do so! Again, because I have access to multiple years of roughly equivalent data in terms of dimensionality of each and what the observations are on, I can also see how well my models trained on some given year, 2014 for example, do at predicting the behavior of those same stocks not just in the following year, 2015, but also potentially how well they can predict the 2016, 2017, and even 2018 behavior as well. Presumably of course, as the years between the training set and the testing set increase, it is less and less plausible that the models will be able to predict accurately or reliably, but it will be a useful exercise nonetheless.

Because I misinterpreted the instructions for Part 2 of our Big Data Analytics Project for this course, I actually began work on carrying out the data wrangling, data-preprocessing, and modeling process in R for all of my Classification and Regression models 3 or 4 weeks ago and just finally realized my error on Monday the 28th. To be sure, I was not able to debug all 12-14 of these models so that they all run and produce output that is of the correct form, but I was able to for most of them by now. As a result, I will share some of my preliminary results in this report.

General remarks regarding the datasets:

1. In terms of the volume of the datasets, on average, there are data on approximately 4k stocks in each of the 5 datasets, and each of them has 225 columns.
2. Some financial indicator values are missing (NaN cells), so the user can select the best technique to clean each dataset (drop.na, fill.na, etc.).
3. There are outliers, meaning extreme values that are probably caused by mistypings. Also in this case, the user can choose how to clean each dataset (have a look at the 1% - 99% percentile values).
4. The third-to-last column, Sector, lists the sector of each stock. Indeed, in the US stock market each company is part of a sector that classifies it in a macro-area. Since all the sectors have been collected (Basic Materials, Communication Services, Consumer Cyclical, Consumer Defensive, Energy, Financial Services, Healthcare, Industrial, Real Estate, Technology and Utilities), the user has the option to perform per-sector analyses and comparisons.
5. The second-to-last column, PRICE VAR [%], lists the percent price variation of each stock for the year. For example, if we consider the dataset 2015_Financial_Data.csv, we will have:
    - 200+ financial indicators for the year 2015;
    - percent price variation for the year 2016 (meaning from the first trading day on Jan 2016 to the last trading day on Dec 2016).
6. The last column, class, lists a binary classification for each stock, where

- for each stock, if the PRICE VAR [%] value is positive, then class = 1. From the perspective of a hypothetical trader, class = 1 identifies those stocks that he or she should BUY at the start of the year and sell at the end of the year for a profit.
- for each stock, if the PRICE VAR [%] value is negative, then class = 0. So, similarly. from a trading perspective, a class value of 0 identifies those stocks that an hypothetical trader should NOT BUY, since their value will decrease, meaning negative profit (losses).

The fact that both the PRICE VAR [%] column and class column are included makes it possible to use these datasets for both classification and regression tasks:

- If the user wishes to train a machine learning model so that it learns to *classify* those stocks that in buy-worthy and not buy-worthy, it is possible to get the targets from the class column;
- If the user wishes to train a machine learning model so that it learns to *predict* the future value of a stock, it is possible to get the targets from the PRICE VAR [%] column.

As previously indicated, there are, on average, data on about 4k stocks in each of the 5 datasets, and each of them has 225 columns. However, as is more common than not in real world analytics projects, these datasets were nowhere near suitable for analysis in their initial form. And so it was only after a significant amount of data wrangling, munging and preprocessing in which I removed all nominal columns from the dataframes, removed all NAs or interpolated their values (I arbitrarily chose to remove all NAs for the datasets used to train and test all my Classification models and interpolated their values for the training and testing datasets used for all my Regression models), removed all near zero variance candidate predictors, and removed all candidate predictors which have higher than 80% correlation with one or more other candidate predictors (to eliminate any collinearity or multicollinearity), that I ended up with data which I could use to train, tune, validate, and test my forecasting models.

The resulting datasets were of the following dimensions: only 513 observations and only 110 of the original set of 225 candidate predictors left for the 2014 dataset, 597 obs & 110 IVs for the 2015 data, 740 obs & 110 IVs for the 2016 data, 758 obs & 110 IVs for the 2017 data, and 793 observations & 110 candidate predictors (aka Independent Variables) for the 2018 stock market data.

## Results of my Classification Models in Terms of How Well They Classified Stocks Which a Hypothetical Investor in 2014 would be Recommended to Buy by our Models

The *Confusion Matrix* for Logit (513 obs):

| Classification | Increase | Decrease |
|---|---|---|
| Increase | 53 | 301 |
| Decrease | 60 | 183 |

*Accuracy* $= 0.4$ with a 95% CI of (0.36, 0.44)

*Kappa* $= -0.08$ *TPR* $= 0.38$ & *TNR* $= 0.47$

*PPV* $= 0.75$ *AUC* $= 0.601$

*Confusion Matrix* for PLS-DA:

| Classification | Increase | Decrease |
|---|---|---|
| Increase | 50 | 311 |
| Decrease | 63 | 173 |

*Accuracy* $= 0.37$ with a 95% CI of (0.34, 0.41)

*Kappa* $= -0.11$ *TPR* $= 0.36$ & *TNR* $= 0.44$

*PPV* $= 0.73$ *AUC* $= 0.62$

*CF* for the Elastic Net Model:

| Classification | Increase | Decrease |
|---|---|---|
| Increase | 50 | 311 |
| Decrease | 63 | 173 |

*Accuracy* $= 0.27$ with a 95% CI of (0.24, 0.31)
*Kappa* $= -0.05$ *TPR* $= 0.17$ & *TNR* $= 0.72$
*PPV* $= 0.72$ *AUC* $= 0.60$

*CF* for an NN Classification Model:

| Classification | Increase | Decrease |
|---|---|---|
| Increase | 51 | 252 |
| Decrease | 62 | 232 |

*Accuracy* $= 0.47$ with a 95% CI of (0.43, 0.52)
*Kappa* $= -0.04$ *TPR* $= 0.48$ & *TNR* $= 0.45$
*PPV* $= 0.79$ *AUC* $= 0.55$

The *Confusion Matrix* for AvgNNets:

| Classification | Increase | Decrease |
|---|---|---|
| Increase | 53 | 269 |
| Decrease | 60 | 215 |

*Accuracy* $= 0.45$ with a 95% CI of (0.41, 0.49)
*Kappa* $= -0.05$ *TPR* $= 0.44$ & *TNR* $= 0.47$
*PPV* $= 0.78$ *AUC* $= 0.58$

*Confusion Matrix* for an SVM:

| Classification | Increase | Decrease |
|---|---|---|
| Increase | 63 | 351 |
| Decrease | 50 | 133 |

*Accuracy* $= 0.33$ with a 95% CI of (0.29, 0.37)
*Kappa* $= -0.08$ *TPR* $= 0.28$ & *TNR* $= 0.56$
*PPV* $= 0.73$ *AUC* $= 0.61$

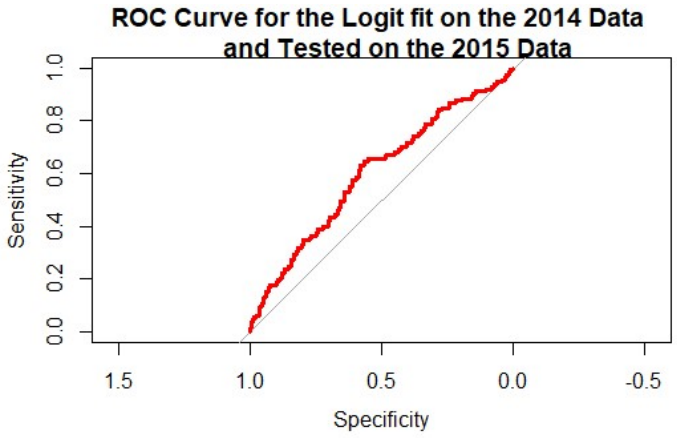
The *CF* for the KNN Model:

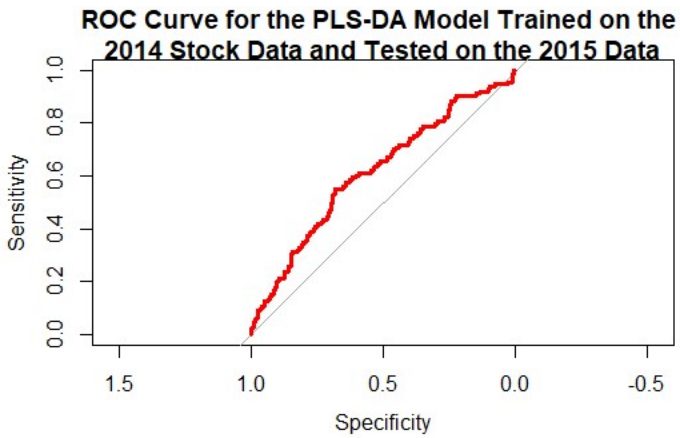
| Classification | Increase | Decrease |
|---|---|---|
| Increase | 85 | 379 |
| Decrease | 28 | 105 |

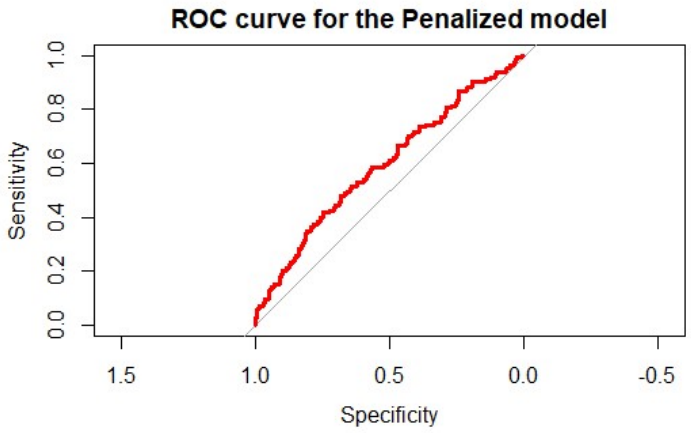
*Accuracy* $= 0.32$ with a 95% CI of (0.28, 0.36)
*Kappa* $= -0.01$ *TPR* $= 0.22$ & *TNR* $= 0.75$
*PPV* $= 0.79$ *AUC* $= 0.53$

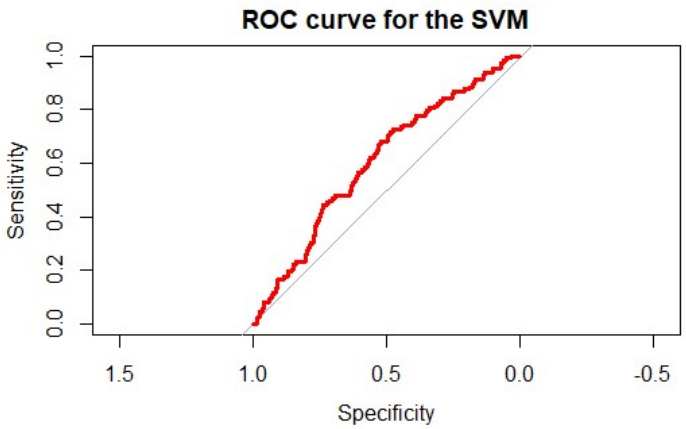
For part 1 of this predictive analytics project, the part concerning classification modeling, the most important data visualizations to include for each of our classification models are their ROC plots.

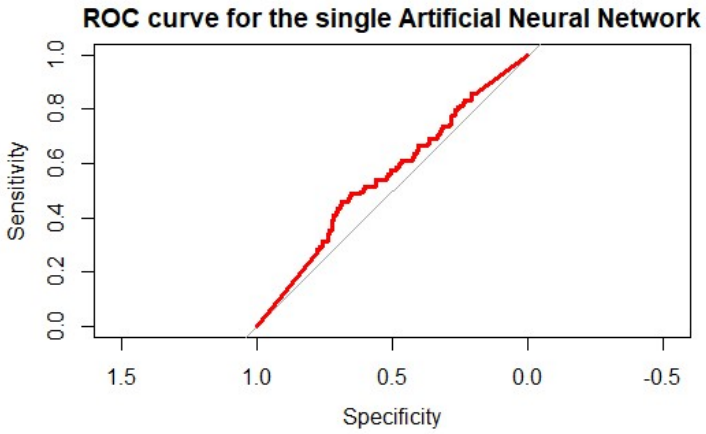
The ROC plots for each of my Classification models are included below:

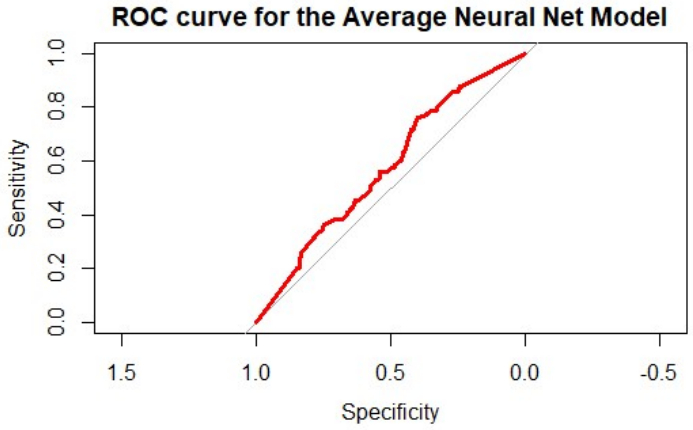


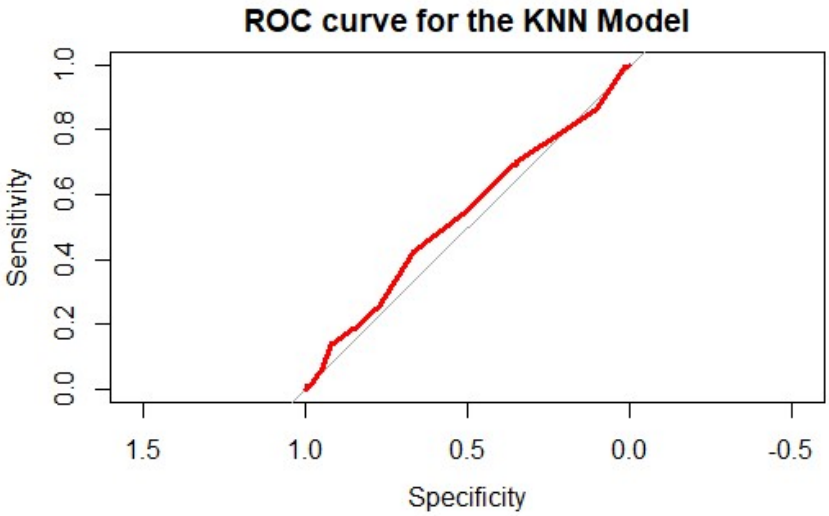




ROC curve for the Penalized model
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.5
1.0
0.5
0.0
-0.5
Specificity


ROC curve for the SVM
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.5
1.0
0.5
0.0
-0.5
Specificity


ROC curve for the single Artificial Neural Network
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.5
1.0
0.5
0.0
-0.5
Specificity


ROC curve for the Average Neural Net Model
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.5
1.0
0.5
0.0
-0.5
Specificity


ROC curve for the KNN Model
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.5
1.0
0.5
0.0
-0.5
Specificity

**References**

*200+ Financial Indicators of US stocks (2014-2018)*. (n.d.). www.kaggle.com. https://www.kaggle.com/datasets/cnic92/200-financial-indicators-of-us-stocks-20142018

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning* (G. Casella, S. Fienberg, & I. Olkin, Eds.; 8th ed.). Springer.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.