Shareholder Class Action Litigation Risk Summary Report

William Terpstra

Data Science Capstone

Dr. Andrew Banasiewicz

5/30/22

Table of Contents

Executive Dashboard

According to predictive models…

The likelihood of shareholder class action litigation against Mohawk Industries, Inc. over a four year period is 9.6%

A 95% confidence interval of the cost of a shareholder class action settlement against Mohawk industries ranges from $4,990,437 to $63,655,296.

The corresponding risk capital breakeven point would occur when the premium of a D&O insurance policy with the adequate $63 million in coverage would be less than $1,527,727 annually.

Introduction

Among the ecosystem of executive threats, shareholder class action (SCA) litigation requires special attention. SCA litigation occurs when shareholders believe they have been misled by a company about its current or future financial prospects and sue to recoup their purported losses from the drop in stock across the period of deception. This perception of deception is a risk inherent in the separation of a company's ownership and management, which results in "conditions of divergent interests and asymmetric information" that can lead to "non-conformance of managers' behaviors to shareholders' expectations of those behaviors" (Banasiewicz, 2015). The threat of SCA litigation requires special attention because it is "one of the most reputationally and economically damaging risks confronting directors and officers of public companies" (Banasiewicz, 2015). Beyond these economic consequences the interrelated reputational damage is equally if not more devastating. Reputation after all is one of a company's most precious and difficult to accrue intangible assets, affecting the future decisions of investors, consumers, and employees alike. Consequently, mitigating the risk exposure of SCA litigation is necessary through the purchasing of Directors' and Officers' insurance. This begs the question, what is the optimal amount of insurance to buy to provide adequate coverage with minimal premiums? The purpose of this report is to provide data-derived estimates of Mohawk Industry's likelihood of SCA litigation and corresponding settlement severity in order to enhance decision maker's D&O insurance procurement decisions.

Data and Approach

The data used in this analysis consists of four sets of data used for the predictor variables, one set of data used for the response variables, and a data dictionary that provides the variable definitions for the preceding data sets. The four data sets used for the predictor variables include fundamentals, a data set of financial performance data reported annually by public companies via SEC form 10-k; stocks, a data set of company stock specific details; securities, a data set providing additional company stock information; and ratings, a file which includes company credit ratings. The data set for the response variable contains all companies that have faced SCA litigation over a ten year period and the corresponding settlement amounts if applicable.

The analytic goal of this report as previously mentioned is the creation of a data derived likelihood and severity estimate of SCA litigation for Mohawk Industries. This requires the creation of two distinct machine learning models. In both cases supervised learning will be used since this problem involves training models with labeled data to predict outcomes. In addition, for each estimate several machine algorithms will be compared to ensure a relatively high performance model. Machine learning algorithms have an accuracy-interpretability trade off. In this model use-case, accuracy is incredibly important since there is a big cost to inaccurately predicting the likelihood or severity of SCA litigation. However, comparison of generally more accurate models with generally more interpretable models is still worthwhile since if the difference in performance is negligible, it would make sense to choose a simpler and more interpretable algorithm. All data cleaning, manipulation, model creation, and final model

prediction was done in the R programming language using the Rstudio integrated development environment. Models were built and tuned using the tidymodels framework. Data pre-processing wasn't only contained to a tidymodels recipe however, and some preprocessing steps like variable transformations were done before the final data set was assembled.

Data Cleaning and Feature Engineering

Standard Procedure Overview

All the raw data used in this analysis required comprehensive data cleaning procedures to be usable. In general for each data set:

- Duplicate observations were removed
- Variables were reduced to only the unique variables of interest
- Variable types were assessed and corrected
- The date range of the data set was filtered to fiscal year 2010-2014
- The date range of the data set was collapsed by summarizing numeric variables as their mean and standard deviation across the date range and by summarizing categorical variables as their mode across the date range
- Numeric variables had their outliers capped at the 95th percentile
- Variables had their distributions assessed for potential imputation issues
    - Numeric variables were transformed if there was a compelling reason to do so (log transforms were used if relative changes in the given variable were more informative than absolute ones)
- New Features were created if relevant
- Variables that had over 20% of values missing were removed

Afterwards the final data set was assembled and several additional data pre-processing steps were conducted:

- Missing values were imputed using MissForest (a robust, nonparametric method that can handle continuous and categorical data)
- The Boruta algorithm was used as a all relevant feature selection technique
- Numeric variables were z-score normalized (standardization)
- Categorical variables were one hot encoded
- Highly correlated variables were removed

The specifics of this process for each data set are discussed below.

SCA Settlements Data Cleaning and Feature Engineering

The main issue with the SCA settlements data set was the lack of gvkey, a variable that serves as a unique identifier for each company and would be used in assembling the final data set used to train and validate models. In order to address this, all predictor data sets were searched

for company tickers that appeared in the SCA settlements data set and if a match was found the gvkey was appended to the relevant observation in the SCA settlements data set. Using this technique, a gvkey was able to be matched for 90 percent of companies that appeared in the SCA settlements data set. This seems like a very good rate of finding corresponding gvkeys given the fact the SCA settlements data spans fiscal year 2005 to 2015 while all other data sets only cover the 2010 to 2014 time period. Consequently it is plausible that some of the companies that faced SCA litigation before 2010 are no longer public or no longer exist. A cursory review of a few tickers that were missing corresponding gvkeys corroborates this. Looking up a few of these tickers on the SEC website, Audible Inc. (ADBL) went private in 2008, Able Laboratories Inc. (ABRX) was dissolved following bankruptcy in 2006, and Northwest Airlines (NWAC) was absorbed into Delta Airlines in a merger in 2008. After adding company gvkeys to this data set, minimal cleaning was left. All that was left to do was reduce this data set to the variables of interest: gvkey, SettlementAmount, and FilingYear then add an additional factor variable named Sued with the value "yes" for all observations and remove duplicate observations. When the SCA settlements data set was eventually merged into the final data set all NAs in the Sued column could be replaced with "No" and Sued would serve as an indicator variable for a company having faced SCA litigation for building classification models.

Ratings Data Cleaning and Feature Engineering

The ratings data set had several problems that needed to be addressed. First of all, the data set's unique variables of interest that provided information about company credit and debt ratings were rife with missing values. Looking at the distribution of missing values on a per company basis revealed that the three variables of interest (splticrm, spsdrm, spsticrm)[1] were missing between 79% to 100% of data values. Since this data set was created from four years of monthly reports, it is possible that this could be the result of data being recorded much more often than companies report. To rule this out the time dimension of this data had to be dealt with. Since the final models are going to predict SCA litigation likelihood and severity on a per company basis, the final data set required one company per observation. All data sets used in this analysis span a date range however, so this date range would have to be standardized and then compressed. To do this the standard procedure was to use the dplyr package group_by() and summarize() functions to transform the data set so each row would correspond to one company. First all duplicate observations were removed. Then all numeric variables would be summarized as two new variables: their mean and standard deviation across the date range. All nominal categorical variables would be summarized as their mode across the date range. Finally, all ordinal categorical variables would first be transformed into a numeric scale and then summarized the same way as numeric variables. This technique for compressing the data into a one company per observation format was implemented on the ratings data set, with credit rating variables first transformed into a numeric scale because they were ordinal categorical variables. Afterward, the percent of missing values was checked again to determine if the missingness was

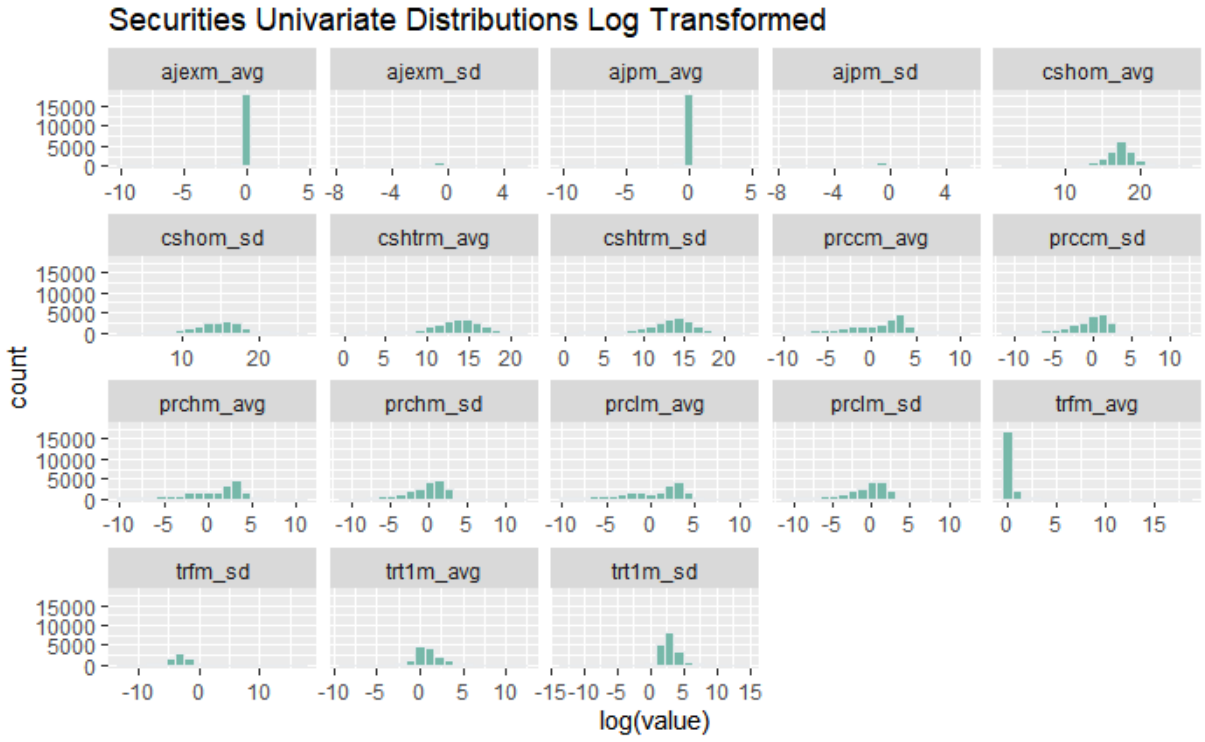
[1] Referring to S&P Domestic Long Term Issuer Credit Rating, S&P Subordinated Debt Rating, and S&P Domestic Short Term Issuer Credit Rating respectively.

the result of much more frequent data collection than reports, but unfortunately this was not the case with 70% of values still missing when each observation corresponded to one company. While missing value imputation was conducted after the final data set was created, this amount of missing data is so high that the informative value of these variables of interest was unfortunately severely compromised.

Securities Data Cleaning and Feature Engineering

For securities data cleaning, first the variables of interest were selected which were all variables that weren't company demographic information besides gvkey. Next variable data types were reviewed and corrected. Datadate was converted into a "POSIXct" date and time object to assess the data date range and ensure it was the same as the rest of the predictor data sets.

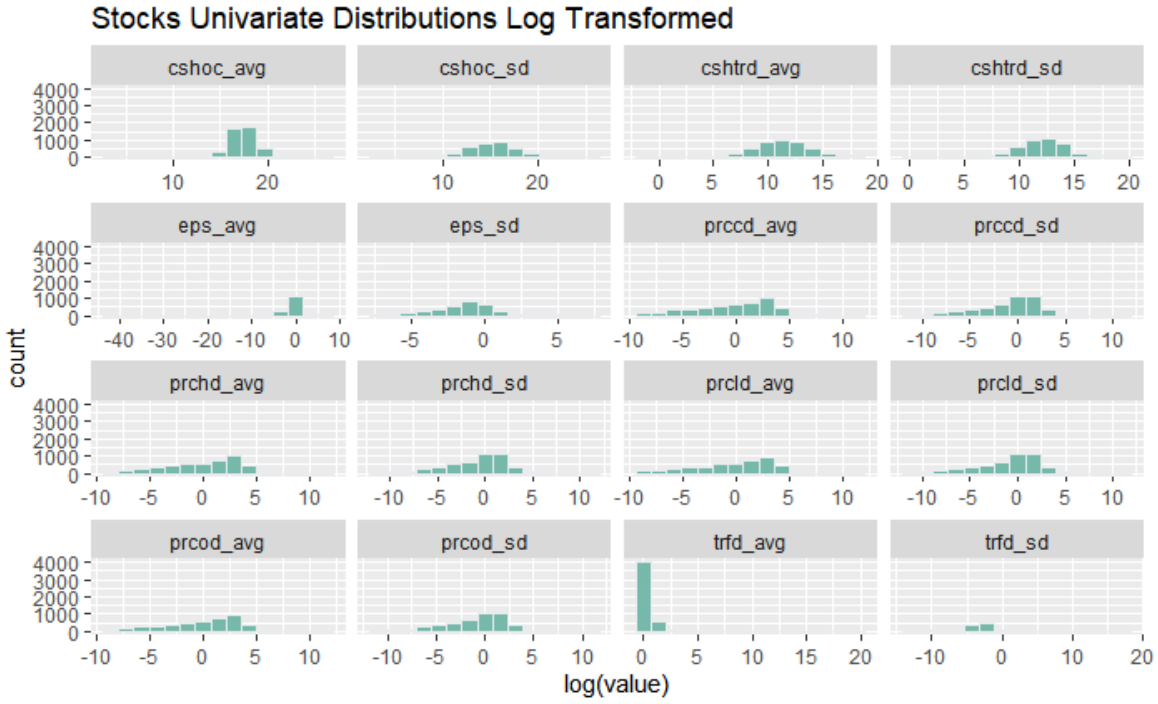
Since this was the first data set involving numeric predictors, the numeric variables had to be assessed for suspicious occurrences and relevant transformations. This was done manually by looking at histograms of each variable because the data set was relatively small. In the end no suspicious distributions were identified however for several variables log transformations seemed appropriate. Specifically cshtrm, prchm, and trt1m were log transformed since it not only addressed skew, which could benefit both linear and tree based model performance, but more importantly relative changes in these variables seemed more relevant than absolute changes. In regards to likelihood models for instance, relative changes in stock price at close (prchm) or trading volume monthly (cshtrm) captured by a log transformation seemed more informative than absolute changes in regards to SCA litigation.



At this point, once again the data was compressed from across the 2010-2014 date range by summarizing the data on a per company basis by taking the mean and standard deviation of numeric variables and the mode of categorical variables. Afterward one new feature was created: average monthly share turnover, which was created by dividing the average monthly trading volume by the average monthly shares outstanding. Missing values were then assessed and variables with more than 20% missing values were removed. Finally, all outliers were capped using the industry standard 95% winsorization method.

Stocks Data Cleaning and Feature Engineering

Manipulation of the stocks data set began once again with the removal of all irrelevant columns like company demographics. Next the data date range was assessed to ensure it was the same standard range as other predictor data sets. Afterward several variables misidentified as numeric variables were properly set to the categorical data type (stko, gind, ggroup) and duplicate observations were removed. Numeric predictors then had their distributions assessed for potential imputation errors or transformations. The variable cshtrd, common shares traded daily, had a suspicious amount of zeros, but upon closer inspection of the data it was plausible that some stocks simply don't have shares traded as often as this data set had observations recorded. Two monetary variables, eps and prchd, were found to be good candidates for log transformation, and were log transformed to reduce skew and emphasize relative over absolute changes. Next, the data was once again collapsed across the 2010-2014 date range using the process described in the securities data cleaning and feature engineering section. Afterward average share turnover was again created as a new feature for this data set, though with this data set it was average share turnover daily rather than monthly. Variables with over 20% missing values were removed and all numeric variables were capped at the 95th percentile.



Fundamentals Data Cleaning and Feature Engineering

The first step of data cleaning for the fundamentals data set involved filtering the data to a single format using the datafmt variable. After the data was filtered, duplicate observations were removed and the data date range was checked to ensure it matched the rest of the predictor data sets.

The fundamental file is massive so the easiest way to begin to assess it was to remove all the columns that have too many missing variables to be usable. This should be an appropriate methodology since the data is reported on a fiscal year basis and there generally shouldn't be missing values. The threshold for variable removal was once again 20% of missing values. After culling variables with large amounts of missingness, the data set was culled to a much more manageable 350 variables down from over 2000. This data set was reduced further by removing all company demographic variables. At this point variable types were assessed and corrected (namely several factor variables including stko, idbflag, auop, ceoso, and cfoso). Afterward the data was once again summarized across its date range using the previously described process and outliers were capped. The resulting data set had over 469 variables. While these variables could be each manually researched and selected, it seemed more effective and efficient to include all variables with relatively high data quality in the composite data set and then perform feature selection using several filters and a wrapper method to produce the final data set.

Final Data Set Creation and Preprocessing

Assembling the final data set involved joining all 4 of the cleaned predictor data sets together by gvkey. However since these are disparate data sources there was a subjective determination involved in deciding what kind of join to use. Conducting an outer join between every data table led to the largest possible data set but resulted in a massive amount of missing values across all variables and consequently wasn't viable. An inner join between all data tables on the other hand produced a very small data set that was unlikely to be a representative sample. As a result the final data set was assembled by inner joining the fundamentals and securities data sets while left joining the rest of the data sets. This methodology resulted in a decent amount of observations, a relatively low amount of missing values, privileged the most information dense data set, and privileged the data set that should theoretically have rigorous reporting standards given it is derived from SEC 10-k form filings. Prior to being joined to the final data set, the SCA filings data set was filtered to the same date range as the predictor data sets and then the filing year variable was removed.

At this point final data set creation and processing diverged. For the severity models' final data set, the data was filtered to observations that had a corresponding settlement amount. Each data set also had the other's predictor variable removed.

For both the likelihood and severity data sets the amount of NAs were assessed. Variables with more than 20% of values missing after the joining procedure were removed. The remaining missing values were addressed using MissForest imputation. MissForest imputation was selected as the missing value imputation method because it is a robust imputation method that is

nonparametric, can handle numeric and categorical variables, is computationally inexpensive, can handle high dimensionality, and provides error estimates using OOB error. For the likelihood data set the OOB error estimate for numeric imputation was .35 (NRMSE) while for categorical variable imputation it was .04 (misclassification rate). For the severity data set the OOB error estimate for numeric imputation was higher at .37 but the OOB error rate for categorical variables was .07. After this step the observation for Mohawk Industries with missing data imputed was saved for each data set to be used in final model predictions later then removed from each data set so it was not used in model training.

To reduce dimensionality the Boruta feature selection algorithm was then used on each data set. Boruta is an all relevant feature selection method making it ideal for identifying all variables of interest from these relatively high dimension data sets. After running the Boruta algorithm on both data sets, the dimensionality of the likelihood data set was reduced to a reasonable observation to predictor ratio while the severity data set still had high dimensionality. This was a consequence of the fact the amount of observations with settlement amounts in the 2010-2014 period was very limited. As a result, all the algorithms selected for severity modeling had embedded feature selection in an attempt to further address this.

Finally, a tidymodels preprocessing recipe was created for each data set. This recipe would be applied before data was used in predictive models and implemented some final data preprocessing steps including the normalization of all numeric variables, the one hot encoding of all nominal categorical variables, the removal of all highly correlated variables, and the removal of all zero variance variables.

```
```{r Likelihood Generic Preprocessing}
#Generic tidymodels recipe for pre-processing likelihood data

#formula for identifying predictors
Data_rec <- recipe(Sued ~ ., data = Data[,features_selected]) %>%
  #updating the variable gvkey to a non-predictor role
  update_role(gvkey, new_role = "ID") %>%
  #normalization so feature scales are commensurate
  step_normalize(all_numeric(), -gvkey) %>%
  #turning nominal factors into dummy variables
  #so they are compatible with certain modeling functions
  step_dummy(all_nominal(), -Sued, -gvkey) %>%
  #smote makes sense for classifiers sometimes
  #but we are looking at likelihood and don't want to mess up the probabilities
  #step_smote(Sued, over_ratio = 1) %>%
  #important to note Boruta does not address colinearity
  step_corr(all_numeric(), threshold = 0.7, method = "spearman") %>%
  #removing all zero variance variables
  step_zv(all_predictors())

#code to assess if the recipe is working properly
test <- Data_rec %>% # use the recipe object
  prep() %>% # perform the recipe on training data
  juice() # extract only the pre-processed data frame
```
```

Likelihood Model Comparison and Findings

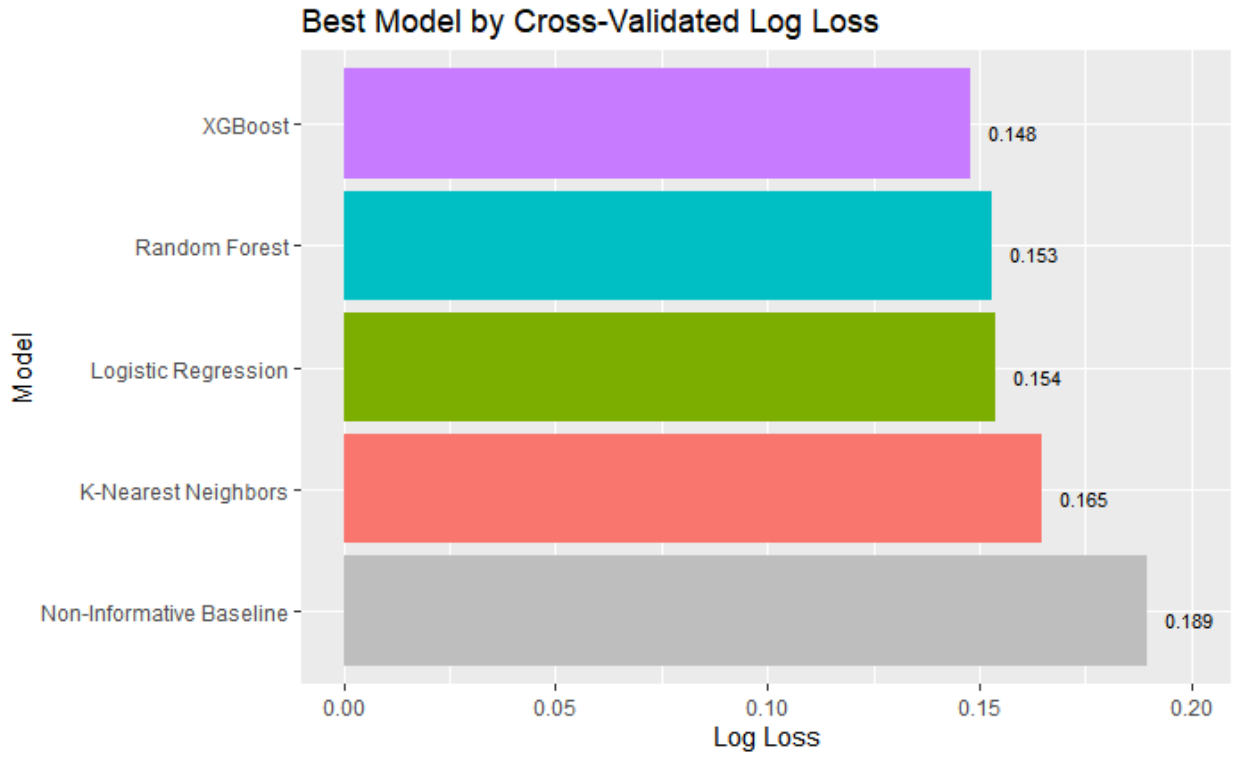
Likelihood Models Evaluation Metric Selection

While the most popular metric for assessing the quality of classification models is accuracy, in this context accuracy is an unreliable measure of model performance because the data set has heavy class imbalance with the positive case of SCA litigation occurring only 4.6% of the time. Consequently, different metrics must be selected to evaluate model performance.

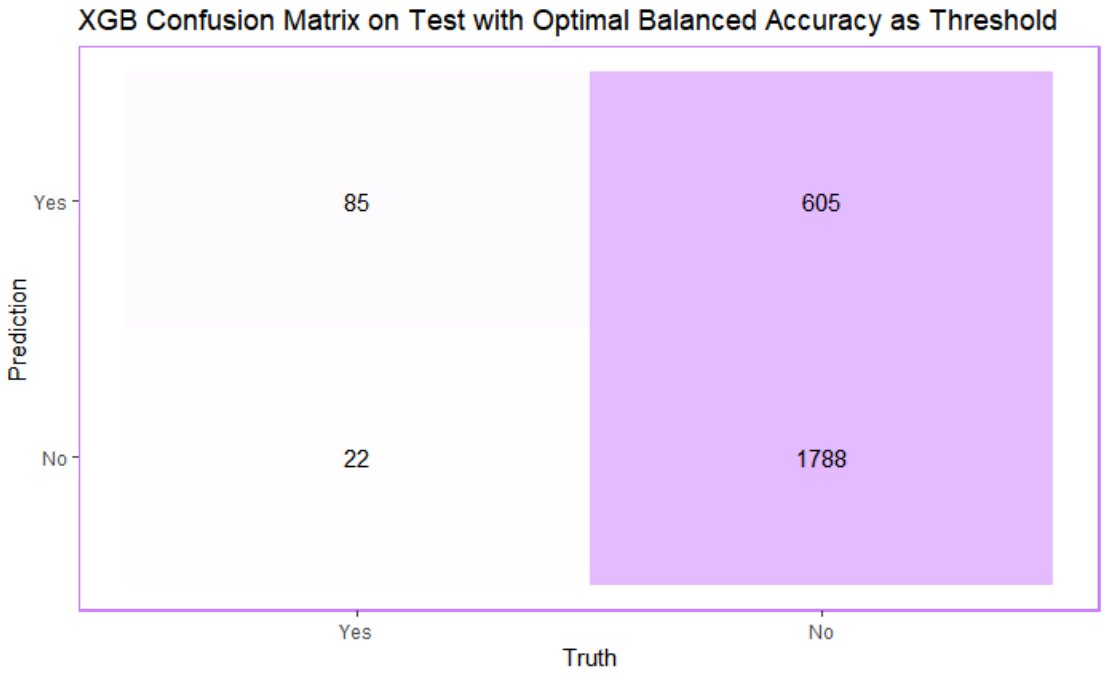
For binary classification there is a plethora of available metrics to evaluate performance. Generally these metrics can be separated into three categories: metrics that assess how well a classifier labels data, like a confusion matrix; metrics that assess how well a classifier discriminates between classes, like AUC; and metrics that assess how well the underlying probabilities used to label the data perform, like log-loss and Brier Score. In the case of likelihood modeling it is most important to optimize the calibration of the prediction probabilities. Consequently, log-loss was used to evaluate model performance and select the optimal model, while the final model's performance was evaluated with a confusion matrix, its log-loss relative to a naive reference model, and most importantly its calibration curve after calibration.

Likelihood Models Validation

To construct the best possible likelihood models, a large number of candidate models across several algorithms were compared. The algorithms chosen were a selection of high performant and more interpretable models and included regularized logistic regression, random forests, XGBoost, and K-nearest neighbors. These hundreds of candidate models were compared using a flat stratified 5 fold cross validation procedure. In this procedure the data set was split into a training set and a testing set in a 75/25 ratio while preserving the distribution of classes. The log-loss of each candidate model was then evaluated using 5 folds stratified cross validation on the training set, with the best model from each algorithm selected for comparison. The best XGBoost model had the lowest log-loss at .148, a significant decrease compared to the .189 non informative baseline for this binary class distribution, and was selected as the optimal likelihood model. This model was then trained on the training set and its final performance was assessed on the held out test set.



Since this was an imbalanced classification problem, a confusion matrix with the default 0.5 decision threshold was non-informative. As a result a more appropriate decision threshold was chosen for generating a confusion matrix on the test set by calculating the threshold that maximized balanced accuracy of the model during the cross-validation procedure used for model selection.



The resulting confusion matrix highlights some key metrics of model performance. The final model achieves a sensitivity of 79.4% and a specificity of 74.7%. While recall is relatively high, precision is very poor at 14%. Thus the final model does a relatively good job of recalling the positive class, whether a company will face SCA litigation, but has a very high false positive rate.

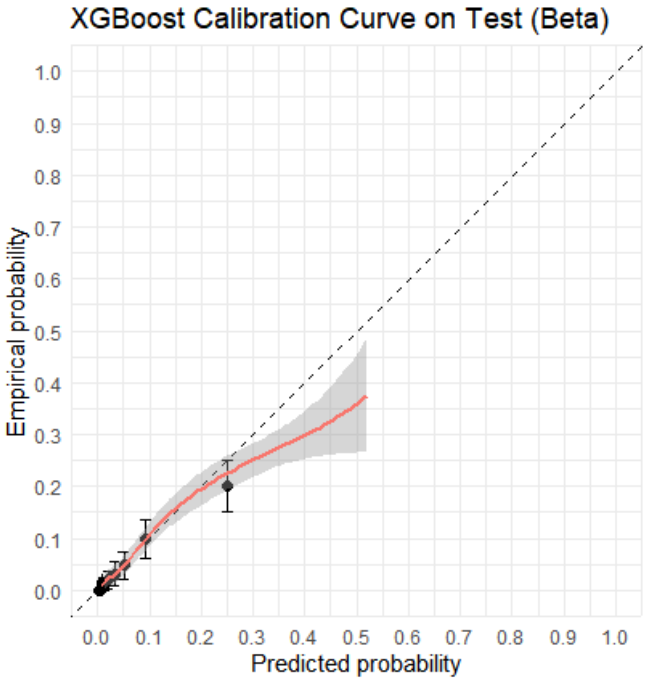
Still, more important than the labels produced at an arbitrary threshold are the quality of the underlying probabilities produced by the model. Since XGBoost does not naturally produce calibrated probabilities, cross validation was used to assess the performance of the base model and ensembles with a variety of calibration techniques. The ensemble of the final XGBoost model with beta calibration performed the best with a log loss of .1468, a 23% reduction in log-loss compared to the naive reference model that always predicted the class distribution.

.metric <chr>	.estimator <chr>	.estimate <dbl>	calibration <chr>
mn_log_loss	binary	0.1894803	reference model
mn_log_loss	binary	0.1489729	none
mn_log_loss	binary	0.1622657	platt scaling
mn_log_loss	binary	0.1468540	beta calibration

4 rows

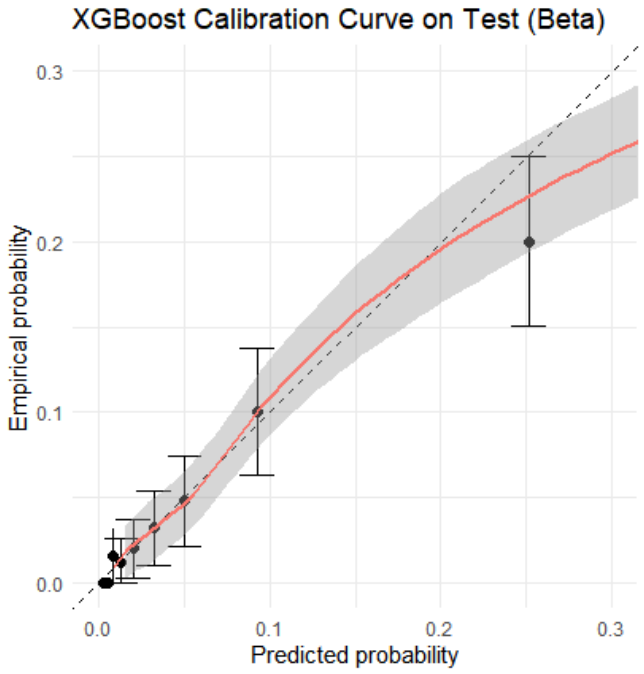
On the test set this beta calibrated model had a final log-loss of .1418, which was even lower than the reference model. Furthermore it has a Brier Score, the MSE equivalent for probability predictions, of .03869.

Assessing the model's calibration curve, the model seems relatively well calibrated. While its probability estimates become significantly conservative at relatively high probabilities, this is not bad considering the extreme class imbalance. Where there is data it is generally well calibrated.



Zooming in on the calibration curve, the calibration issues at extremely low probabilities are more apparent. For extremely low probabilities, the final model is slightly more conservative

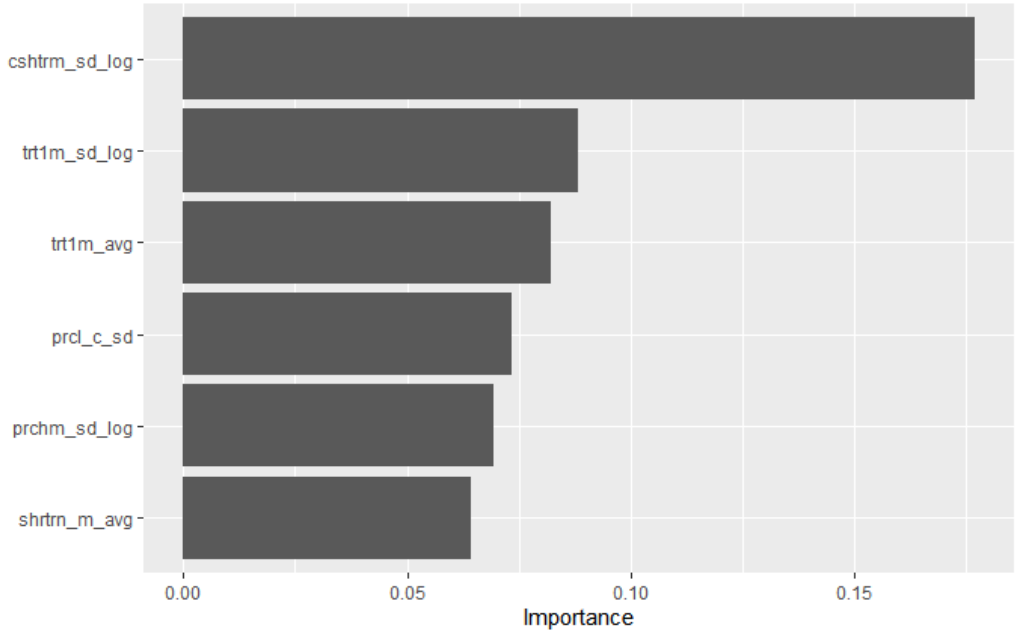
than it should be.



Best Likelihood Model Findings

This final ensemble of the best XGBoost model with beta calibration trained on the training set[2] predicted that Mohawk Industries Inc. had a 9.61% chance of facing SCA litigation over a four year period. In addition to this final likelihood estimation, XGBoost models can also produce a feature importance plot.

[2] Given the likelihood model building procedure, while training the model on all available data for maximum performance is tempting, this would make the final likelihood model's external performance estimate no longer valid.



According to this plot, the five most important variables for the best XGBoost likelihood model were the log of the standard deviation of the total number of shares traded monthly, the log of the standard deviation of the monthly total return of a given company, the average monthly total return, the standard deviation of the low of share price per calendar year, and the log of the standard deviation of the monthly high of share price. The sixth most important variable was average share turnover monthly, a variable created during feature engineering.

Severity Model Comparison and Findings

Severity Models Evaluation Metric Selection

To compare severity models, evaluation metric selection was much more straightforward. Common metrics include root mean squared error (RMSE), mean absolute error (MAE), and the coefficient of determination (R^2). RMSE and MAE both have their use cases but for the purposes of this report RMSE was used for model comparison since it penalizes large prediction errors more than MAE, which is prudent when it comes to severity estimation. In addition R^2 was also used for goodness of fit evaluation.

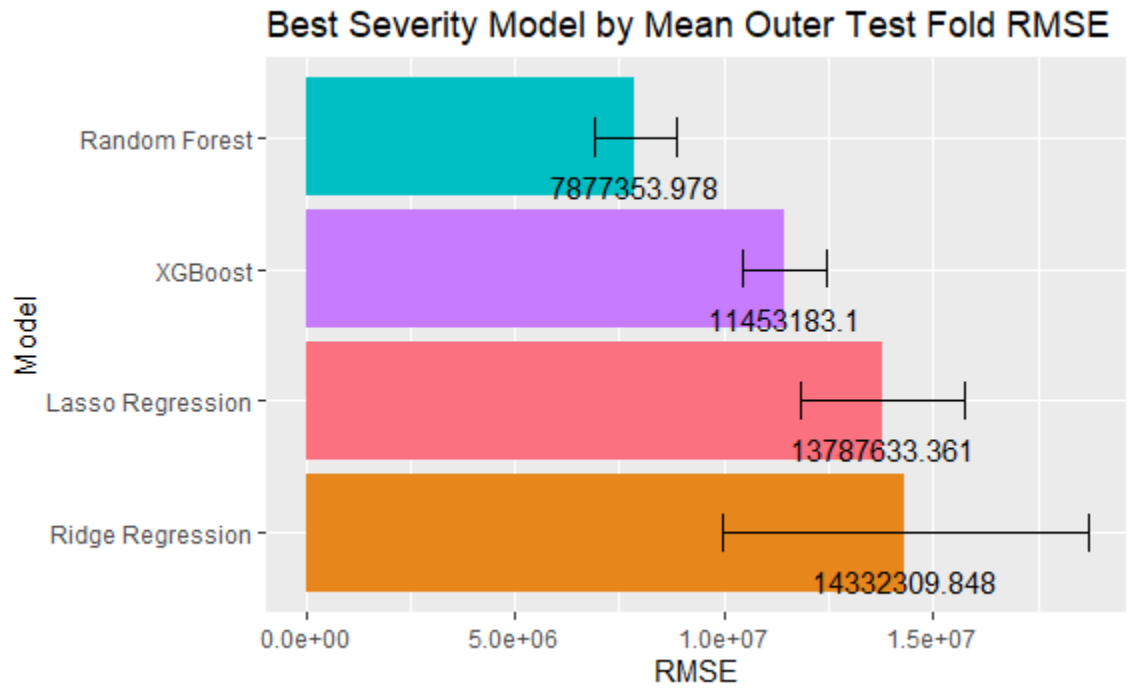
Severity Models Validation

The creation of a high quality severity model necessitated optimizing model hyperparameters and comparing multiple algorithms. The algorithms selected for comparison were a combination of high performance models (random forest regression and xgboost regression) and more interpretable models (lasso regression and ridge regression). In order to optimize hyperparameters and compare algorithms a nested cross validation procedure was implemented since the data set was relatively small with only 59 observations, thus utilizing the

data budget optimally was essential. Nested cross-validation entails "nesting" a cross-validation loop, referred to as the inner loop, for hyperparameter optimization within another cross-validation loop, referred to as the outer loop, for estimating generalization accuracy. This procedure for model comparison optimally navigates the training and test data split trade-off[3] that is especially detrimental to small data sets and circumvents the optimization bias introduced by using the same k-fold cross validation procedure to both tune model hyperparameters and evaluate models. In this case the Raschka nested cross-validation method was used, which entails having 5 outer folds and 2 inner folds. The 5 models created at the end of the outer loop of cross validation then had their performance averaged to provide the unbiased algorithm performance estimate and the standard deviation of this estimate is calculated as well to provide a measure of variation. Afterward the mean R^2 was also calculated.

model <chr>	rmse <dbl>	std <dbl>	r2 <dbl>
Random Forest	7877354	1124776	0.7379249
XGBoost	11453183	1136663	0.5311328
Lasso Regression	13787633	2226692	0.3932631
Ridge Regression	14332310	4983485	0.5215982

4 rows



[3] Specifically maximizing model performance entails increasing the amount of training data available at the cost of the test data available which decreases model generalization performance. Having a small data set makes navigating this problem difficult.

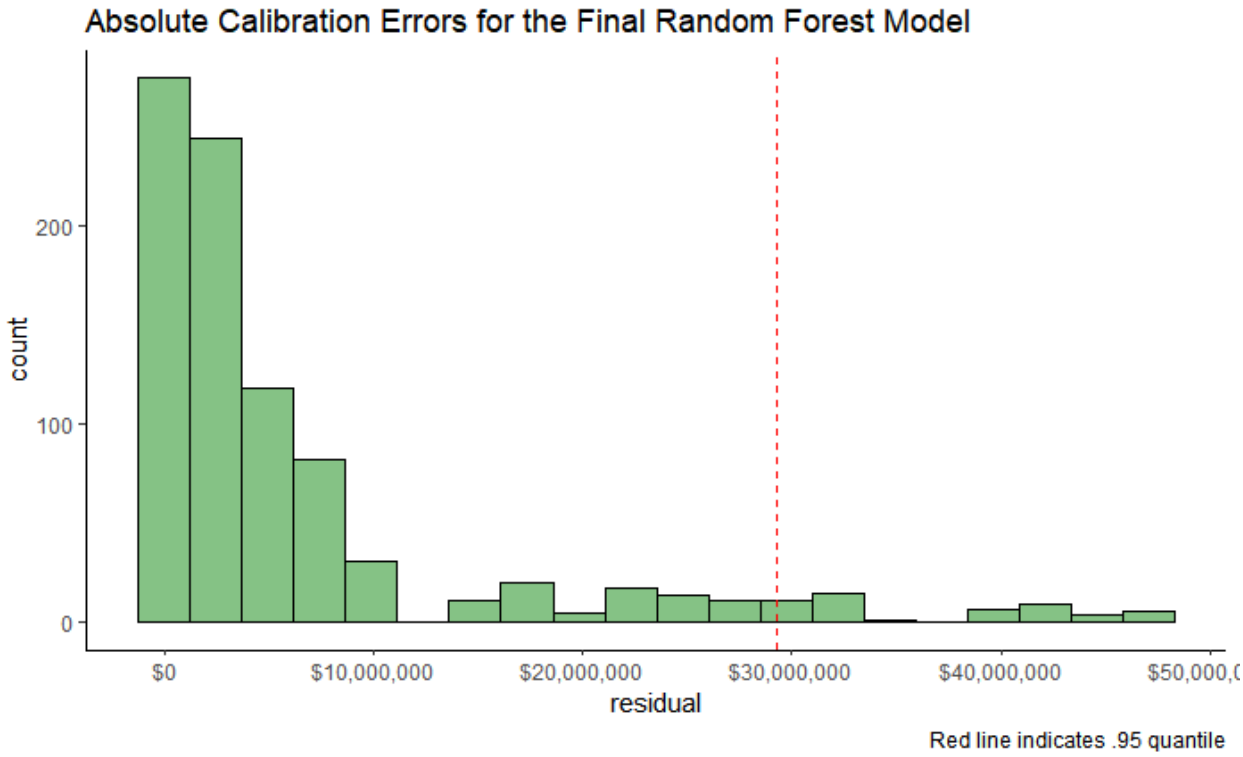
From these results it is clear random forest is the most high performance algorithm. Not only does it have the lowest RMSE, but also the error bars constructed from 95% confidence intervals of the standard deviation don't overlap with the error bars of any other algorithm. As for random forest's unbiased estimate of goodness of fit, the mean R^2 was .73 which is a good R^2 for financial modeling. Interpreting this R^2 can be a bit tricky since nested-cross validation is a way to estimate unbiased generalization performance, not the performance of a single model. Thus the mean R^2 of .73 indicates that 73% of the variance in SCA settlement amount is explained by the predictors in the final data set with a hyperparameter optimized Random Forest regression approach. At this point the final severity model is created in the Raschka nested cross-validation procedure by tuning a random forest model's hyperparameters using a cross validation procedure on the whole data set, and then training the best model. The whole data set can be used in final model tuning and training since the nested cross-validation procedure already produced unbiased estimates of generalization performance.

Best Severity Model Findings

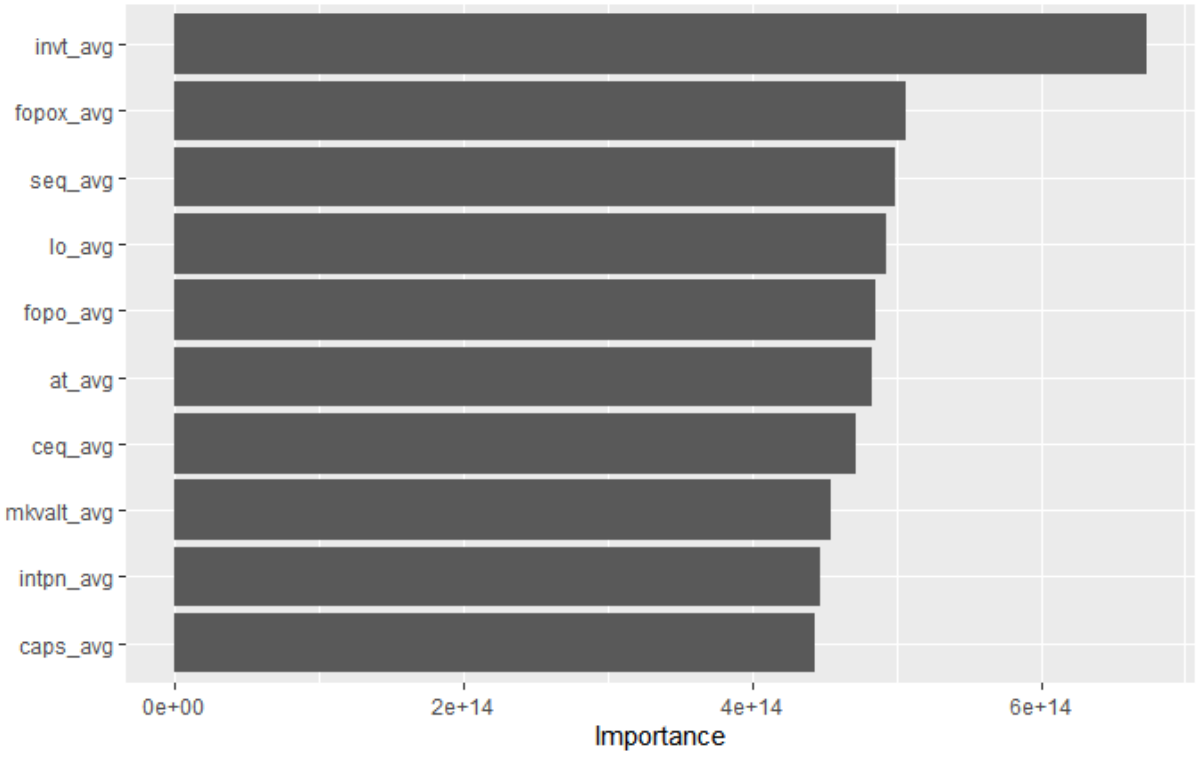
Predicting a point estimate of SCA litigation using the final random forest model was straightforward, and resulted in a predicted settlement amount of $34,322,867 for Mohawk Industries. However while a point estimate is useful, a prediction interval would be much more informative for decision makers. A prediction interval is a range of values that quantifies the uncertainty of a prediction for a given observation, much like how a confidence interval quantifies the uncertainty of a population parameter estimate. Since prediction intervals are constructed from a single observation, they inherently have more uncertainty than confidence intervals.

While calculating prediction intervals is straightforward for valid linear models, predictions for other kinds of models are more complex. In this case naive prediction intervals were constructed using a conformal inference method, specifically the method implemented in the MAPIE Python library.[4] MAPIE stands for Model Agnostic Prediction Interval Estimator and entails fitting the model on training data, making predictions on a test data set, computing the absolute errors, calculating the 1 minus alpha quantile from the distribution of these errors, and finally creating the interval by adding then subtracting this quantile from a given prediction. Using this method a 95% prediction interval for Mohawk Industries was constructed, ranging from $4,990,437 to $63,655,296.

[4] An alternative approach would involve quantile regression forests.



In addition to the severity estimate prediction, the final random forest model can also be used to generate a feature importance plot to identify the most important variables. While not as interpretable as regression coefficients, the variable importance plot highlights the most important features for the final severity model.



According to the plot, the top five most important features are: the average total inventory value, the average funds for operations in the other category excluding tax benefit, the average stockholders equity of the parent company, the total of liabilities in the other category, and the funds from operations in the other category. Since the model was relatively good, further investigation of these variables is warranted for additional insights pertaining to their role in explaining the variance of SCA settlement amounts.

Recommendations

As discussed earlier, the risk of SCA litigation requires special attention, but it is ultimately only one part of the ecosystem of executive threats. Consequently how the final likelihood prediction and severity model prediction interval is used is best up to the discretion of the decision makers responsible for managing the executive threats facing Mohawk Industries, who have a full picture of the situation. Still, a recommendation for insurance procurement can be made based on the risk-capital breakeven point where the potential cost of a realized risk like SCA litigation is equal to the cost of an insurance premium for a policy with sufficient coverage. Based on the model estimates, the probability that SCA litigation will occur in a given year is 9.6% while the top end of the 95% prediction interval is $63,655,296. Consequently at the risk capital breakeven point the cost of an adequate D&O insurance policy's annual premium would be $1,527,727.

Limitations

There are several notable limitations in this analysis. First, the model selection process for likelihood model creation has a chance of overfitting. Specifically, since cross-validation is used to both optimize model hyperparameters and evaluate performance for model selection there is a chance the model performance estimate is optimistically biased. While this is not a problem for measuring absolute performance, since a held out test set is reserved for estimating generalization performance, this may be problematic for measuring relative performance between models. If this optimistic bias varies between models, there is a chance it could disrupt the true performance rankings of the models and lead to suboptimal model selection. This issue could be avoided by implementing a nested cross-validation approach as used in severity model creation, however this would also be exponentially more computationally expensive.

Second, potential data leakage is also an issue in this analysis. While variable normalization is implemented in a way as to not leak information, the Boruta feature selection procedure was not. Ideally Boruta feature selection would be nested within the model development process and run for each fold in the cross-validation process. However, again this would lead to an exponential increase in computational resources required and as a result was not currently a viable approach.

Tidymodels Specifications and Hyperparameter Tuning Grids

The following code excerpts contain the details of how each model's specifications were set up using the tidymodel's framework, which hyperparameters were tuned for each model, and how the corresponding hyperparameter grids were created for use with the tune_grid() function in the tune R package. Following the validation procedures outlined in this report, the conclusions should be reproducible with these details. The full code used to produce the models and visuals in this report is available at the following github repository: https://github.com/wtwillterp/Data_Science_Capstone

Likelihood Models

```
#regularized logistic regression likelihood model specifications
lr_spec <- logistic_reg(penalty = tune(), mixture = 1) %>%
        #important to set family to binomial since this is a classification problem
        set_engine("glmnet", family="binomial")

#grid for creating logistic regression candidate models
lr_grid <- tibble(penalty = 10^seq(3, -4, length.out = 40))

#random forest likelihood model specifications
rf_spec <- rand_forest(mtry = tune(), min_n = tune()) %>%
        set_engine("randomForest", num.threads = parallel::detectCores()) %>%
        set_mode("classification")

#grid for creating the random forest likelihood candidate models
rf_sv_grid <- 25 #"An integer denotes the number of candidate parameter sets to be created
automatically"

#XGBoost likelihood model specifications
xgb_spec <- boost_tree(trees = 1000,
        tree_depth = tune(),
        min_n = tune(),
        loss_reduction = tune(),
        sample_size = tune(), mtry = tune(),
        learn_rate = tune()) %>%
        set_engine("xgboost") %>%
        set_mode("classification")

#grid design that attempts to maximize parameter space coverage
#used for creating the XGBoost likelihood candidate models
xgb_grid <- grid_latin_hypercube(tree_depth(),
        min_n(),
        loss_reduction(),
        sample_size = sample_prop(),
        finalize(mtry(), Data_train),
```

```
        learn_rate(),
        size = 40)

#K-nearest neighbors likelihood model specifications

knn_spec <-
        nearest_neighbor(neighbors = tune()) %>%
        set_engine("kknn") %>%
        set_mode("classification")

#grid for creating the knn likelihood candidate models
knn_grid <- tibble(neighbors = c(1:29, seq(from = 30, to = 100, by = 10), seq(from = 120, to =
200, by = 20)))
```

Severity Models

```
#random forest regression specifications for severity modeling
rf_sv_spec <- rand_forest(mtry = tune(), min_n = tune(), trees = 1000) %>%
        set_engine("ranger",
        num.threads = parallel::detectCores(),
        importance = "impurity") %>%
        set_mode("regression")

#random forest grid for optimal hyperparameter search in the inner loop
rf_sv_grid <- 25 #“An integer denotes the number of candidate parameter sets to be created
automatically”

#lasso regression specifications for severity modeling
lassor_spec <- linear_reg(penalty = tune(), mixture = 1) %>%
        set_engine("glmnet")

#lasso penalty grid for optimal hyperparameter search in the inner loop
lassor_grid <- tibble(penalty = 10^seq(8, -8, by = -.25))

#ridge regression specifications for severity modeling
ridger_spec <- linear_reg(penalty = tune(), mixture = 0) %>%
        set_engine("glmnet")

#ridge penalty grid for optimal hyperparameter search in the inner loop
ridger_grid <- tibble(penalty = 10^seq(8, -8, by = -.25))

#XGBoost regression specifications for severity modeling
xgb_sv_spec <- boost_tree(trees = 1000,
        tree_depth = tune(),
        min_n = tune(),
        loss_reduction = tune(),
        sample_size = tune(), mtry = tune(), ## randomness
        learn_rate = tune(), ## step size) %>%
```

```
        set_engine("xgboost") %>%
        set_mode("regression")

#XGBoost grid design for optimal hyperparameter search in the inner loop
#latin hypercube attempts to cover high dimensional parameter space efficiently
xgb_sv_grid <- grid_latin_hypercube(tree_depth(),
        min_n(),
        loss_reduction(),
        sample_size = sample_prop(),
        finalize(mtry(), Data_train),
        learn_rate(),
        size = 40)
```

References

Banasiewicz, A.D. (2015) 'Quantifying Executive Threats: Shareholder Litigation', *International Journal of Business Competition and Growth*, Vol. 4, No. 1/2.

Banasiewicz, A. D. (2015). The ecosystem of executive threats: A conceptual overview. *Risk Management*, *17*(2), 109–143. http://www.jstor.org/stable/43695459

Raschka, S. (2018). Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *ArXiv, abs/1811.12808*.