Prioritizing Settlement of Workers' Compensation Medical Claims

Kristy Cheng, Megan Muller, Ming Zhang, UCSB
March 2019

Advisors: Ian Duncan FSA FIA FCIA FCA CSPA MAAA
Janet Duncan FCAS FSA MAAA and Xiyue Liao PhD.

Sponsor: John Alltop, CSAC - Excess Insurance Authority

List of Figures

List of Tables

Abstract

Insurance companies have found that claim costs continue to increase the longer a claim is open (partly due to ongoing legal fees). Therefore, it is imperative to handle and settle claims on a timely basis. California State Association of Counties - Excess Insurance Authority (CSAC-EIA) would like us to analyze claim data to determine the criteria for predicting type of claim settlement of workers' compensation medical claims, meaning claims we think have a reasonable probability of being closed via Compromise and Release (C&R) Settlement versus closed by other means.

Background

Workers' compensation was created to protect employees who become injured or ill while on the job. It also protects the dependents of workers who are killed while working. Workers' compensation provides monetary compensation for the injuries or illnesses sustained by the claimant while on the job.[1] Workers' compensation covers, but is not limited to, medical payments, lost wages, and vocational rehabilitation. We will be focusing on the medical aspect of these claims.

C&R settlement is one of the most important claim settlement types, and the claim type our sponsor CSAC-EIA is most interested in. A C&R settlement only occurs when a claimant is permanently disabled, and the percent of disability can be determined and the claimant has reached a stable health condition. This means the claimant is at a point at which maximum medical improvement (MMI) has been reached. Once these requirements have been met, which can take a long time, the process can be negotiated quickly. Negotiations are entered into and once finished, the check for the settlement amount is sent shortly thereafter. The check is required by law to be sent within 30 days of final settlement of the claim. This is a preferred settlement route for the insurer, because when settled this way a claimant takes a lump sum, thus replacing an unknown cost with a known cost. Additionally, the claimant gives up their right to

[1] LII Staff. "Workers Compensation." *LII / Legal Information Institute*, Legal Information Institute, 4 Jan. 2018, www.law.cornell.edu/wex/workers_compensation.

reopen the case, which frees the insurance company from having to pay any future medical costs. [2]

Project Hypothesis/Objective

A faster closed claim is a claim that costs CSAC-EIA less. The sponsor, CSAC-EIA would like the team to create a model which can predict when a claim closure is likely to be a C&R settlement, so they can be monitored and effectively closed if and when it meets the necessary requirements.

Data Description

We received one dataset in two excel files from CSAC-EIA with 1,312,150 rows and 117 variables in total, which are very large and comprehensive. The data included detailed information about each claim such as program year, evaluation date, entity, relevant dates and amounts and so on. We created a data profile to help us understand the data. The data profile lists each variable, the meaning of the variable and the range of values. The data profile is attached to this report as Appendix B. A few key variables are described below:

- Program year indicates the effective period for the policy. Our data has program years ranging from 1967/68 to 2017/18. However, data prior to 1994/95 appears incomplete (and is also less relevant due to its age). Therefore, our analysis will only use data from program years 1994/95 and subsequent.
- Evaluation date is the date when the company looks at the claim, which is June 30th each year. Our data includes evaluation dates ranging from 1996 to 2018. However, for our analysis, we only need the latest evaluation of each claim which reflects the most current information about each claim. Therefore, we will only use the June 30, 2018 claims evaluations.
- Claim type categorizes each claim based on its characteristics, e.g., medical only, permanent disability, temporary disability, etc. There are 10 claim types in our data as

[2] Acevedo, Carli. "What Is a Compromise and Release in California Workers' Compensation?" *Shouse California Law Group*. www.shouselaw.com/workerscomp/compromise-and-release.

shown in the data profile (Appendix A). Several of these claim types would not be subject to potential settlements, e.g., medical only claims are short duration claims which are closed quickly. Our analysis needs to focus on long duration disability claims. Therefore, we will only the following claim types in our analysis: temp disability, future medical, indemnity and death.

- Settlement type describes how the claim settled. The settlement types are shown in the data profile (Exhibit A). In our data, about 80 percent of claims are not settled. The C&R settlement type is the most important settlement to our sponsor.

Data Processing

Combining Datasets

We processed the datasets in R, which is an open-source statistical programming language. The first step in data processing was to combine the two datasets. During this process, a problem we met was the unmatched date format in two datasets. Since the first dataset has no null values in column “Date Entered” whereas the second dataset does have null values, when R automatically read files, it would take the first “Date Entered” column as POSIXt format, and the second one as character format, where dates are written as numbers. So, to combine two datasets, we transferred the characterized numbers into numeric ones, and then formatted them as dates, with origin date set on “1899-12-30”. Last, we combined the two datasets.

Data Cleaning

After we combined two datasets, we tried to prune it for future analysis.

- We only kept the Temporary Disability, Future Medical, Indemnity, and Death only claim types, because those are the claim types that can lead to C&R settlement.
- We deleted claims with Program Years before 1994/95, because they contain a relatively small amount of data and are less representative of current claims.

- We discovered 13 columns that contain a single data value, so we deleted them because they would not give us any useful information during the future prediction.[3]
- We filtered out rows containing null values in the column "Claim Status on 06-30-2018". Our sponsor explained that some clients were moving from CSAC-EIA, so that conducting model on these data serves no purpose for our project.
- Finally, we formed our final dataset by choosing only those claims that have an evaluation date on June 30th, 2018. Otherwise, we are duplicating the same claim over and over again in our dataset and using outdated views of the claim.

After all these steps, we have 35,460 rows and 105 variables in our updated dataset, which is significantly smaller than the original one.

To further create a dataset with only important variables, project advisors suggested a list of variables of interest that possibly contributes to C&R settlements, while some are grouped by either interval, frequency, or logic.[4]

- We kept the columns Gender, Average Weekly Wages, Safety, Claim Type, Settlement Type, Litigation, Paid ALAE Over 5K Flag, Incurred Total.
- Average Weekly Wages
    - Capped the wages at $10,000.
    - Classified them into 5 intervals. (0, 1, 600, 900, 1250, 10000)
- Age at DOL
    - Replaced ages that are less than 18 or greater than 75 years old by NA.
    - Classified them into 5 intervals. (18, 25, 37, 45, 53, 75)
- Years Employed at DOL
    - Replaced negative, and unreasonable (e.g 111 years) values by NA, and change the value <1 to 0.5.
    - Classified into 4 intervals. (0.5, 2, 7, 14, 70)
- Grouped variable into 4 groups with approximately equal sizes.
    - Incurred Total. (0, 1000, 4000, 25000, 1160000)

[3] Columns deleted are listed in Appendix.
[4] See Appendix B for details. Modified variables were renamed.

- Sorted data by frequency, set a cut-off point for the frequency, and any data below that point was grouped into one.
    - Entity Type
    - Class Code
    - Nature of Injury Code
- Body Part Code
    - Grouped manually and logically. (i.e. head/neck/back, arms, legs, trunk, others/multiple body parts, and NAs)
- Cause of Loss Code
    - Grouped manually and logically (i.e. fall, burn, caught, cut/abrasion, vehicle, strain, strike, Other Person/Other/Misc./Act of God, and NAs)
- PD rating
    - Grouped by logic: 0 in one group, all other values in another group, since majority of the ratings are 0s'.
    - Dealt with miscoding (e.g put 181, 400 into the largest group)

Furthermore, new variables were created for better interpretation of the data, especially some information of a claim requires particular computation.

- Closure Year is the number of years between the Date of Loss and Date Closed; Open claims are labeled as NA.
    - Grouped by frequency.
- Reopened Check is an indicator variable, of which 1 is a claim that has been reopened, 0 is a claim that has never been reopened.
- Pension Age is the summation of Age at DOL and Years Employed.
    - Grouped by logic: Under age 50 in one group, 5 years interval after age 50, and over age 80 in one group.
- Age of Claim is the difference between Evaluation Date and DOL, or 6/30/2018 minus Date of Loss.
- Current Claimant's Age is the summation of Age of Claim Age at DOL.
    - Classified into 4 intervals. (0.5, 2, 7, 14, 70)

After all these steps, we have 35,460 rows and 18 variables in our final dataset, ready for data analysis, and modeling.

Model Building

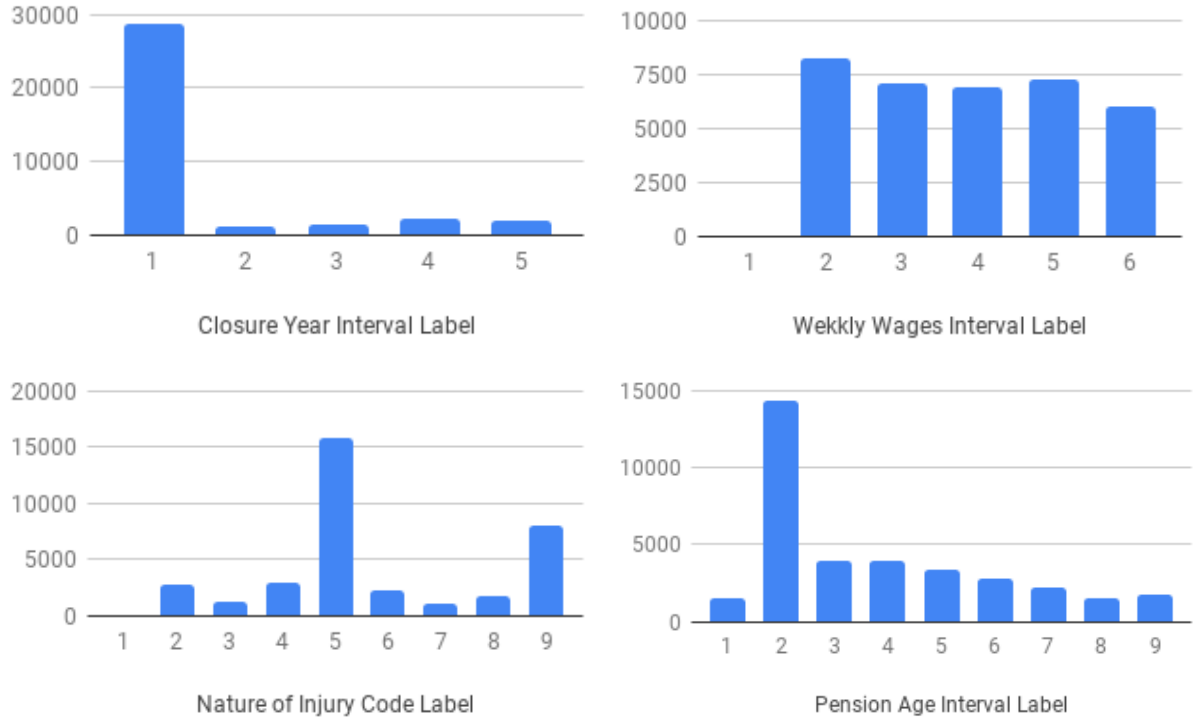
While the goal of the project is to prioritize C&R settlements of claims based on the probability of success, analysis of closed claim data can be used to determine the criteria for predicting prioritization of claim settlements. According to the Department of Industrial Relations in the State of California, "a Compromise and Release Agreement is a settlement which usually permanently closes all aspects of a workers' compensation claim except for vocational rehabilitation benefits, including any provision for future medical care."[5] Since benefits are paid as a lump sum to the claimant, C&R is a cost and time effective process for insurers. Claims with C&R settlement can be permanently closed with no further cost, and liability to the company. Therefore, through the process of exploratory analysis, model fitting, and prediction, the best-fitted model being chosen will prioritize the factors that correlate with C&R settlement.

Exploratory Analysis

Using the processed data, graphs and statistics were generated in order to obtain a general idea of the data, of which any observable and visible patterns, or deviation might imply a higher probability of being a C&R settlement. Hence, a data profile was created to summarize the eighteen variables, which would not only be helpful for referencing, but also imply the distributions of each variable. The histograms in Figure 1 represent the distributions of some variables that we were interested in. For Closure Year, vast majority of the claims has a closure year that is not in the interval, meaning that it is not in [0,43). From the second graph, weekly wages is distributed relatively uniform. For Nature of Injury Code, and Pension age, most claim has a nature of injury of strain and misclassified, and pension age below 50.

[5] "Notice of Options Following Disability Rating," State of California Department of Industrial Relations Division of Workers," Compensation Disability Evaluation Unit, https://www.dir.ca.gov/dwc/DEU110.pdf.

Figure 1: Histograms of Important Variables



Target Variable

A target variable is the response variable predicted in supervised learning, which is represented as a label. A target variable, in this case, is defined as the predicted value of interest with the binary approach based on type of settlement, which focuses on C&R settlements versus All Other settlements. This method allows the team to predict whether a claim is likely to result in a C&R settlement or not based on the information given with indicator values of 1 and 0 separately.

Training and Testing Datasets

Given the variables in the dataset as input, the supervised learning approach was employed to construct predictive models. In order to predict settlement type based on the data, training and testing datasets were being chosen randomly from the final dataset, of which 80% of the data was training data, and the remaining 20% was testing data without replacement. The purposes of

separating the data are to fit models using only the training data, and further evaluate the model using the holdout test data, avoiding the possibility of overfitting.

Data Balancing for Training Dataset

Prior to building model for the training dataset, the dataset had to be balanced in order to construct a useful and unbiased model, since there are only 2,790 C&R settlements out of more than 35,000 observations. To avoid the overwhelmed classification rules due to the prevalent class (All Other), the Random Over-Sampling Examples (ROSE) method from the *ROSE* package, a smoothed bootstrap approach, was used to realize the issue.[6]

Traditionally, the conventional bootstrap method either oversamples the infrequent class by randomly resampling with replacement, while it can also under-sample the prevalent class by randomly dropping the prevalent class, to obtain a balanced dataset. However, since only 7.87% of the data is of C&R settlement, such traditional method creates bias, especially the balanced dataset has to be large enough for modeling. Instead of generating exactly the same data points, the ROSE strategy generates new artificial samples by using a distribution that takes the location of a real point as mean, along with a covariance matrix H.

Before actually implementing the ROSE method, all the NA values in the data had to be relabeled as either "Not in Interval," or "Misclassified," which could be treated as one of the levels; all of the new variable names have the word ".Label" added to the end.[7]

- Original columns with NA: Closure Year Interval, Weekly Wages Interval, Age at DOL interval, Years Employed at DOL, Years Employed at DOL Interval, Class Code, Cause of Loss Code, Nature of Injury Code, Body Part Code, Incurred Interval.

As a result, the balanced training dataset now has 14330 observations for C&R settlement, and 14038 observations for All Other settlements. The test data remains untouched.

[6] Lunardon, N., Menardi, G., & Torelli, N. (2014, June)."ROSE: A package for Binary Imbalanced Learning." *The R Journal, 6/1*.

[7] See Appendix B: Definition of Labeling for details.

Model Fitting

1. Regression Analysis

1a. Logistic Regression: Binomial

One model the team considered for the modeling of C&R claims was logistic regression. Logistic regression is a type of generalized linear model, or a model that extends linear regression to non-normal data. This type of regression works only for models with binary response variables, while multinomial regression is its multinomial counterpart. For logistic regression our goal is to estimate:

$$P(X = j) = f(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p) \qquad (1)$$

where j = 0, or 1 for the binomial case. That is to estimate the probability of a label, Y, given a set of predictors, X, using an appropriate function of X. An appropriate function is a function that maps from the real numbers to a probability interval monotonically. Logistic regression specifically uses the logit function, which models the log odds of a function. The logit function models as follows:

$$logit(p) = ln\left(\frac{p}{1-p}\right) \qquad (2)$$

$$P(Y = j|X) = \frac{e^{\beta' X}}{1+e^{\beta' X}} \quad (3)$$

where $\beta' = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$.

Using the function *glm()* in R with the option *family = "binomial"*, a logistic regression that models settlement type as the response variable against all the other variables in the dataset was built.

1b. Penalized Regression

Shrinkage methods, such as lasso regression and elastic net, are algorithms that automatically shrink the coefficients of insignificant predictors or set them to zero. Such regression analysis method not only serves as a variable selection procedure, but also shows collinearity of coefficient estimates.

Lasso Regularization

"Ridge and lasso regularization work by adding a penalty term to the log likelihood function. In the case of ridge regression, the penalty term is β_1^2 and in the case of lasso, it is $|\beta_1|$."[8] Such regularization methods are essentially feature selection techniques for reducing overfitting.

Elastic Net

Linearly combining the L_1 and L_2 penalty terms of the lasso and ridge regression, respectively, elastic net has a penalty function $\|\beta\|_1 = \Sigma^p_{j=1} |\beta_j|$. Elastic net is able to deal with correlated predictors in data by selecting or dropping those predictors together.

k-fold cross-validation was then performed using the built-in function in R for cross validation to evaluate the reduced models, finding the best penalty parameter, λ, for the model.

Figure 2: Lasso, Ridge, and Elastic Net[9]

$$\text{Lasso:} \quad \hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1$$

$$\text{Ridge:} \quad \hat{\beta}_{\text{ridge}} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda_2 \|\beta\|_2^2$$

$$\text{Elastic Net:} \quad \hat{\beta}_{\text{enet}} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$$

where $\|\beta\|_2 = \sum_{i=1}^{p} \beta_i^2$ and $\|\beta\|_1 = \sum_{i=1}^{p} |\beta_i|$

2. Machine Learning

Aside from regression analysis, machine learning is also an applicable method for predictive modeling. While such an approach is common for statistical decision making, decision trees, random forests, bootstrap aggregating ("Bagging"), boosting, and support vector machine (SVM) are the modeling techniques that the team focused on because the goal of the project is to identify and classify C&R settlements.

[8] K. "A Gentle Introduction to Logistic Regression and Lasso Regularization using R." *Eight to Late*. July 11, 2017 (10:00 p.m.), https://eight2late.wordpress.com/2017/07/11/a-gentle-introduction-to-logistic-regression-and-lasso-regularisation-using-r/.

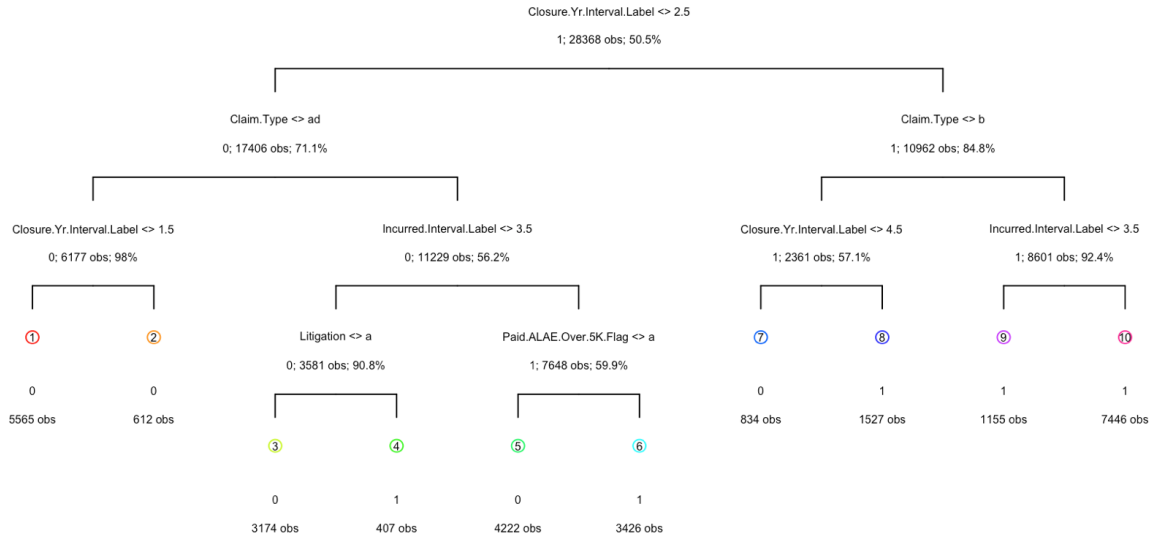
[9] Franks, Alexander. "Regularization." PSTAT 231. June 1, 2018, University of California at Santa Barbara.

2a. Tree Models

Decision Tree

A decision tree is a predictive model that segments the space that is being worked in, into the purest segments possible, by creating rectangles with the maximum amount of a single label possible[10]. For the decision tree, the dataset will be fitted to the tree and form a classification tree, of which the predicted outcome will be the class that the input data belongs to. Figure 3 shows a visualization of the random forest fitted to the data with decision bounds of each node being annotated.

Figure 3: Decision Tree



Bootstrap Aggregating ("Bagging")

As a part of ensemble learning, the bootstrap aggregating, or bagging, method creates and combine multiple classification models to further improve the performance of the prediction. All eighteen predictors from the data are used for each tree, while each is fitted to the bootstrapped training dataset. The bagged decision is then created by "averaging" these large trees through majority-voting to lower the variance, so that the final prediction is consensus among all the trees.

[10] Franks, Alexander. "Decision Trees." PSTAT 231. 2018, University of California at Santa Barbara.

Random Forest

Incorporating all the variations in the data, decision tree and bagged tree tend to split the ends extensively, even though the tree can be partially trimmed by pruning.[11] In contrast with the complexity of the resulting decision tree, random forest is a better method than decision tree, since it avoids the issue of overfitting.

Random forest is a type of ensemble learning, which is a form of machine learning where many classification models are created and combine in an effort to improve the predictive power of the model. [12] In the case of random forest, you use many decision trees to create the best model. The split variable for each split of a single decision tree is chosen from a random sample of p/3 predictors at each split, where p equals to the total number of predictors in training data; 6 random predictors are used for each split in this case. Such method reduces the correlation effect of the bootstrap trees, significantly lowering the variance than bagging does.

Random forest constructs a set of decision trees using the training set, and the output classification of the settlement type is computed as the "average" prediction of the individual trees. Therefore, by creating many of these trees, it is possible to create a model that takes the label predicted by consensus among the separate trees in the forest.

Boosting

Another approach of ensemble learning is the boosting algorithm. For boosting, each fitted model is built from the information and results from the previous fit, aggregating the predictors to attain a better performance (strong learner) than that of a single model (weak learner). Using the *gbm()* function in the gbm package, boosted classification trees could be fitted to the training data with the option *distribution = "bernoulli"* since we have a binary classification.

[11] K. "A Gentle Introduction to Random Forests using R." Eight to Late. September 20, 2016 (9:44 p.m.), https://eight2late.wordpress.com/2016/09/20/a-gentle-introduction-to-random-forests-using-r/.
[12] Franks, Alexander. "The Bootstrap and Ensemble Learning." PSTAT 231. 2018, University of California at Santa Barbara.

2b. Support Vector Machine (SVM)

Support vector machine (SVM) is an algorithm used for binary classification through separating hyperplane with decision boundaries. It is really effective when numbers of variables and observations are small. However, we still want to implement this method to see if it works well. By using Cross Validation to find parameter c (cost), we build this model through *svm()* function by setting linear kernel.

Model Comparison

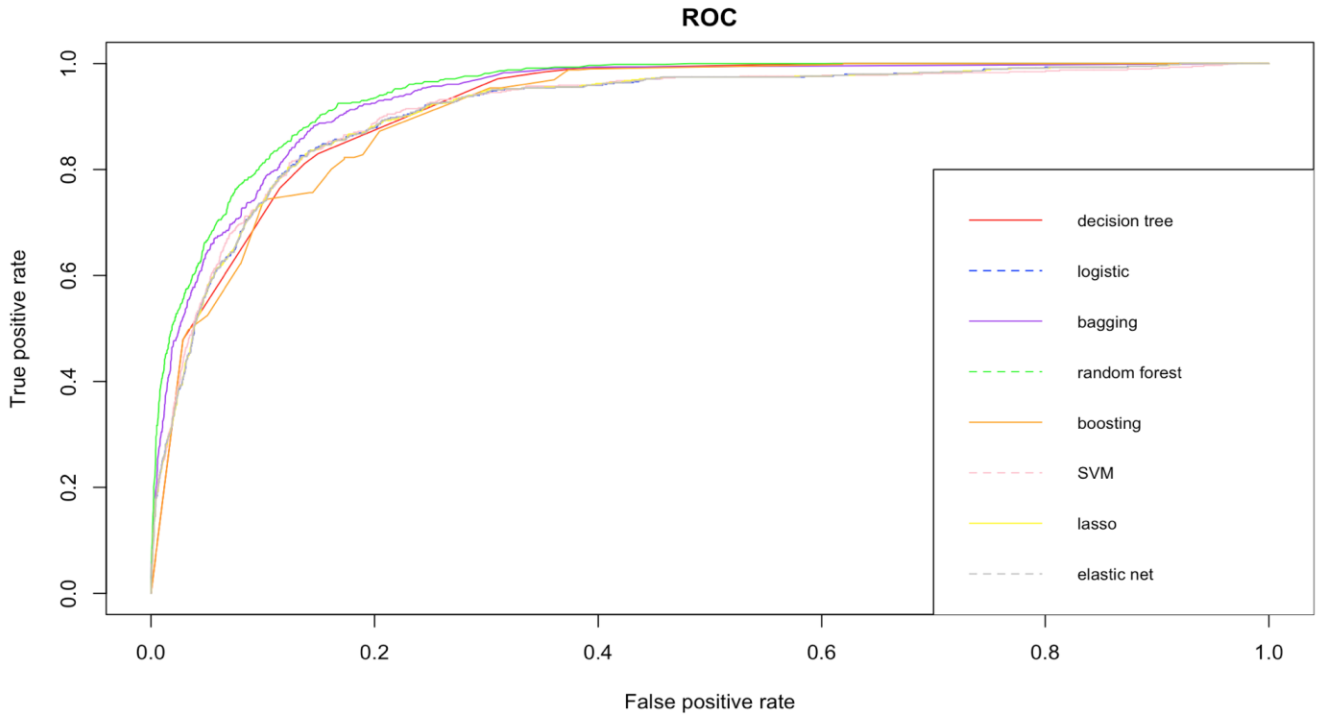
Model Selection

Following the fitting of data, and model building, multiple plausible were evaluated and compared using Receiver Operating Characteristic (ROC) curve, and confusion matrix.

The eight models were assessed through receiver operating characteristic (ROC) curve, and area under the curve (AUC), aiming for a higher true positive rate than false positive rate. This probability threshold is simply a tuning parameter, a p-value, that tells you whether your hypothesized label is true or not. A receiver operating characteristic (ROC) curve is a curve that plots the true positive rate (TPR) vs. the false positive rate (FPR) for each possible p_{thresh}, where p_{thresh} is the probability threshold, a tuning parameter, that tells you whether your hypothesized label is true or not. A TPR is the rate at which the model correctly classifies as a "positive" response, and is calculated $\text{TPR}= \frac{True\ Positives}{True\ Positives + False\ Negatives}$. While the FPR is the rate at which model incorrectly classifies as a "positive" response, and is calculated $\text{FPR}= \frac{False\ Positives}{False\ Positives + True\ Negatives}$.

By examining the ROC curves, shown in Figure 4, the best model is random forest, since its' ROC is the closest to the upper left limit, resulting in the highest AUC value.

Figure 4: ROC Curves



Alternatively, for statistical classification, a confusion matrix can provide a clear visualization of the performance of a supervised learning algorithm. An NxN matrix, for N classes, has rows representing the predicted class from the model, columns representing the actual class from the dataset. In our study, binomial logistic regression corresponds to N = 2, as shown in Table 1. By evaluating the sensitivity, specificity, and accuracy of the model, model should tradeoff and balance these values, considering the "cost" of the false predictions.

Table 1: Confusion Matrix when N = 2[13]

[13]Srivastava, Tavish. "7 Important Model Evaluation Error Metric Everyone should Know," *Analytics Vidhya*. Last modified February 19, 2016. https://www.analyticsvidhya.com/blog/2016/02/7-important-model-evaluation-error-metrics/.

Confusion Matrix		Target		
		All Other	Compromise & Release	
Model	All Other	True Negative (TN)	False Negative (FN)	Negative Predictive Rate (NPR) = TN/(TN+FN)
	Compromise & Release	False Positive (FP)	True Positive (TP)	Positive Predictive Rate (PPR) = TP/(TP+FP)
		False Positive Rate (FPR) = FP/(FP+TN)	True Positive Rate (TPR) = TP/(TP+FN)	Accuracy = (TP+TN)/(TP+TN+FP+FN)

From the confusion, the training error and test error of each model were also derived. For training error, the classifier is trained on the training set, and predictions are made on the training set. Similarly, test error also requires the classifier to be trained on the training set, but predictions are made on the test set. Then, from each of the confusion matrices, both training error rate and test error rate can be calculated as $1 - \frac{(TP+TN)}{(TP+TN+FP+FN)}$.

Figure 5 represents the error rates of the eight models we built, and random forest has the lowest test error of 0.0781, and second lowest train error of 0.008. Hence, random forest has the highest accuracy of predicting classification of settlement type.

Figure 5: Training Error and Test Error

	train.error	test.error
decision tree	0.1589467005	0.15101523
logistic regression	0.1555273548	0.16003948
bagging	0.0002467569	0.08573040
random forest	0.0007755217	0.07811619
boosting	0.1716370558	0.19811055
SVM	0.1536238015	0.15143824
lasso regression	0.1555626058	0.15919346
elastic net	0.1553158488	0.16018049

Now, considering that there is only about 9% C&R settlement in the test dataset, a model can easily have an accuracy of 91% by classifying each claim as "All Other," and missing all the "C&R." However, CSAC-EIA is rather interested in claims to be predicted as "C&R," since claims that are classified as C&R settlement can be passed onto the claim closure specialist to further evaluate. Therefore, a high accuracy, or low test error might not be the most relevant criteria in this case.

In order to capture more C&R settlements, we lowered the FNR, and higher the FPR by selecting a common PPR of about 33%, that is, about one third C&R classifications are correct, and comparing the FNR of each model. A lower FNR was in favor, as FNR is the missed-classified portion of C&R settlement. As shown in Table 2, Random forest remains to be the best model because it has the lowest FNR of 0.085 out of the eight models.

Table 2: FNR with PPR = 33%.

Model	False Negative Rate (FNR)
Decision Tree	0.170
Logistic Regression	0.160
Bagging	0.111
Random Forest	0.085
Boosting	0.244
Support Vector Machine	0.162
Lasso Regression	0.157
Elastic Net	0.158

After computing the metrics above, the performances of the eight fitted models were being examined, and compared. The best-fitting model that best predicts the classification of a given claim is random forest.

Inference

Using the best fitting model, the project also aimed to understand the importance of covariates, and to interpret each variable in order to infer the implied set of criteria that is being used to prioritize settlement activity. Ultimately, the model should be able to not only predict and classify future claims for prioritization, but also infer the relationship between each variable and C&R settlement.

Random Forest Model

In terms of predicting claims with C&R settlement, the confusion matrix shows a clear illustration of the prediction. In Table 3, out of a total of 7092 claims, only 50 claims are misclassified as all other settlements, when they are actually C&R settlement. In other words, the prediction captures more than 90% of the claims with C&R settlement.

Table 3: Confusion Matrix for Random Forest

Confusion Matrix		Target	
		All Other	Compromise & Release
Model	All Other	5435	50
	Compromise & Release	1070	537

To further interpret the random forest, two variable importance plots were built using the function *varImpPlot()* in R, by measuring the impurity decrease during each split using mean decrease in accuracy, and mean decrease in Gini. For the mean decrease accuracy, "The more the accuracy of the random forest decreases due to the exclusion (or permutation) of a single variable, the more important the variable is deemed."[14] The mean decrease in Gini measures the contribution to the homogeneity of the nodes and leaves in the random forest of each variable. For both measures, a larger value implies greater importance of the variable. From Figure 6, and Figure 7, Claim type, and closure year interval are the most important variables in both plots. However, a variable has the largest impurity reduction may still not be the most important, as impurity reduction can only measure how good the split is, instead of directly telling us how good the variable is in modelling. Also, we can see that by using different impurity measure (Accuracy and Gini), we get different variable importance ranks. So, although random forest model gives us the best prediction, it can tell us more details about how the prediction works.

Figure 6: Variable Importance Plot for Random Forest (Mean Decrease in Accuracy)

[14] "Random Forests," Metagenomics. Statistics.. https://dinsdalelab.sdsu.edu/metag.stats/code/randomforest.html.

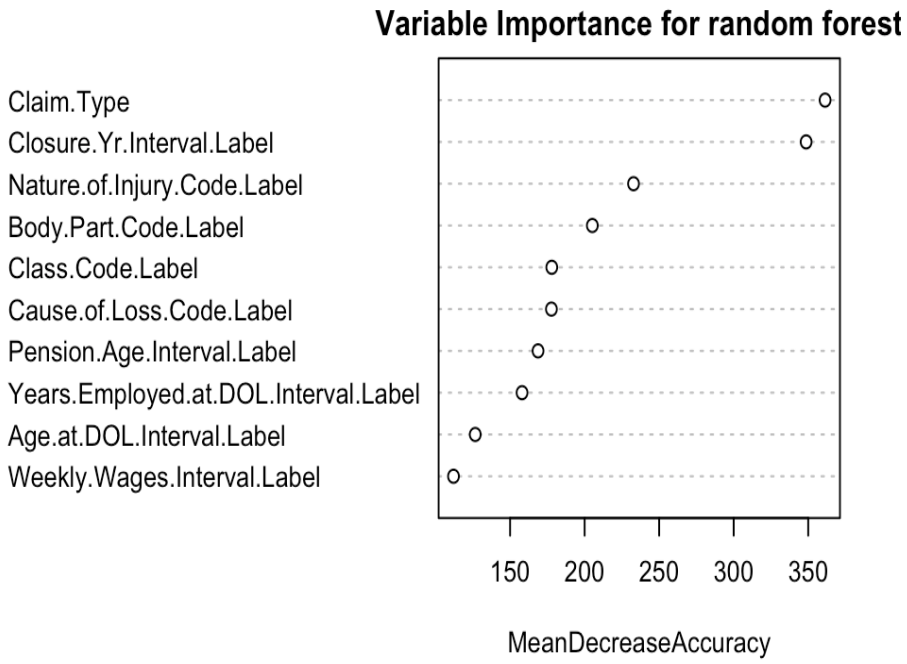
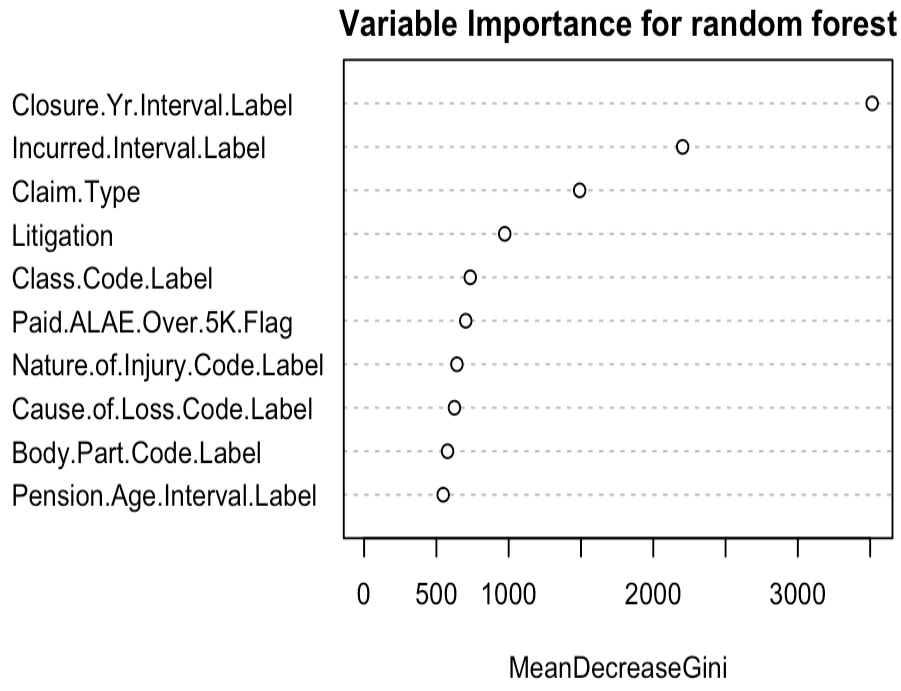


Figure 7: Variable Importance Plot for Random Forest (Mean Decrease in Gini)



Logistic Regression Model

To explore the importance of each variable in detail, we might also want to consider some alternative model that has other advantages than random forest. Even though random forest is the best for classifying settlement type, it can be relatively difficult to interpret. Also, understanding the specific criteria that contributes to C&R settlement can be helpful for the sponsor to further evaluate a claim, after it is being classified as C&R settlement. On method to analyze the data in such aspect is logistic regression.

As discussed in earlier sections, in order to build an appropriate regression model, penalized regression was used to check for multicollinearity. "Multicollinearity (or collinearity for short) occurs when two or more independent variables in the model are approximately determined by a linear combination of other independent variables in the model."[15] High collinearity, in turn, causes the estimated parameters to be unstable; even a minor change in predictors will result in a big change in the estimated parameters. By comparing the FNR of regression models in Table 4, since false negative rates of the three models are fairly close, logistic regression is able to give a more extensive interpretation of the data. A full logistic regression model was chosen because it provides a more insightful model for interpretability, as opposed to a reduced model from penalized regression.

Table 4: FNRs for Logistic Regression, Lasso, and Elastic Net

Model	**FNR**
Logistic Regression	0.160
Lasso	0.157

[15] "Lesson 3 Logistic Regression Diagnostics," UCLA Institute for DIgital Research and Education. https://stats.idre.ucla.edu/stata/webbooks/logistic/chapter3/lesson-3-logistic-regression-diagnostics/.

Elastic Net	0.158

Using the logistic regression model, we were able to examine the level of importance of each variable, as well as each level of those variables. For example, a significant p-value, which has a relatively low value, might indicate higher level of significance. Table 5 summarizes some specific levels of the variables that have one of the highest rankings of importance. Since most levels in Closure Year Interval were significant, the result was consistent with the random forest. Also, Other, Fire, and Police have a higher level of importance in Class Code. After comparing the ranking of p-values for logistic regression model to variable importance plot for random forest, we also found that Future Medical is the most important level among Claim Type, contributing to the overall importance of Claim Type in the model (Table 6).

Table 5: p-value of Variables[16]

Variable (level/label within the variable)	**p-value**
Entity Type Group (Special Districts)	$< 2*10^{-16}$
Litigation (1)	$< 2*10^{-16}$
Paid ALAE Over 5K Flag (1)	$< 2*10^{-16}$
Class Code (7: Other)	$< 2*10^{-16}$
Closure Year Interval (2,3,4,5)	$< 2*10^{-16}$
Incurred Interval (2,3,4)	$< 2*10^{-16}$
Incurred Interval (5)	$3.54*10^{-13}$
Class Code (2: Fire)	$5.08*10^{-13}$
Gender (Male)	$7.68*10^{-12}$
Body Part (6: Trunk)	$3.11*10^{-8}$

[16] See Appendix C: Output for details.

Class Code (3: Police)	$5.45*10^{-5}$

Table 6: p-value of Claim Type

Claim Type	**p-value**
Future Medical	0.0113
Temporary Disability	0.7300
Indemnity	0.2177

Given that the logistic regression model has a response of *log(Odds) = logit(p)*, the relationship between the outcome variable and predictors can also be inferred by the coefficients of the predictors. For instance, Male in the variable Gender(Gender = Male) has a coefficient of -0.28905. By holding all other predictors at constant, the odds of being a C&R claim for males (Male = 1) over the odds of being a C&R claim for not male (Male = 0) is $e^{-0.28905} = 0.7490$. In other words, the odds for male are 25.10% lower (0.7490 - 1 = -0.2510) than the odds for not male of being a C&R claim. Similar concept applies to other predictors in the model, while such inference can provide a more practical outlook and insight of both the model and the data.

Conclusion

Using the processed data, random forest is the best-fitted model. We set the positive predictive rate as 33%, which directly yields to a threshold of 0.21 for this model, and the resulting prediction captures more than 90% of the claims with C&R. With the forest model, a new set of data can now be fed to the predictive model by treating it as the test data. Logistic regression model was also used for its' interpretability because it is also important for us to know the criteria for the prioritization of claims. Future Medical is the most important value in Claim Type, all intervals in Closure Year Interval are significant, and Other, Fire, and Police are the top three levels among class code. Based on the information from logistic regression, insurance company, or claim closure specialist can further evaluate a claim after it is being classified as C&R settlement by the random forest.

Recommendations for Future Studies

The logistic regression model can be improved by incorporating interaction terms. Since our data has a large amount of variables and levels, interaction terms will generate an overly complex model, which is time-consuming to build and to interpret. Quantile regression can also be implemented to examine the effect of predictors on settlement type, given a specific quantile of the dependent variable. Moreover, target variable can be changed from binary outcome to multivariate outcome that looks at C&R settlement (1) vs. other settlements (2) vs. not settled claim (0) using multinomial logistic regression. Having a response variable with more than two outcomes will provide a more informative classification of the claim.

Currently, the prediction probabilities for the boosting method are centered too much, meaning that they have a very low variance. Considering that running boosting and SVM are time-consuming, SPARK may be used for programming, instead of R, to find better parameters. Nevertheless, other sampling approach other than the ROSE method such as, Bayesian nonparametric weighted sampling, may also be utilized to more accurately balance the train data.

References

Acevedo, Carli. “What Is a Compromise and Release in California Workers’ Compensation?” *Shouse California Law Group*. www.shouselaw.com/workerscomp/compromise-and-release.

Boston University School of Public Health. “Regression Diagnostics.” *Correlation and Regression with R*. Last modified January 6, 2016. http://sphweb.bumc.bu.edu/otlt/MPH- Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression7.html.

Franks, Alexander. “Decision Trees.” PSTAT 231. 2018, University of California, Santa Barbara.

Franks, Alexander. “Logistic Regression.” PSTAT 231. 2018, University of California, Santa Barbara.

Franks, Alexander. “Regularization.” PSTAT 231. June 1, 2018, University of California, Santa Barbara.

Franks, Alexander. “The Bias-Variance Tradeoff.” PSTAT 231. 2018, University of California, Santa Barbara.

Franks, Alexander. “The Bootstrap and Ensemble Learning.” PSTAT 231. 2018, University of California, Santa Barbara.

K. “A Gentle Introduction to Logistic Regression and Lasso Regularization using R.” *Eight to Late*. July 11, 2017 (10:00 p.m.), https://eight2late.wordpress.com/2017/07/11/a-gentle- introduction-to-logistic-regression-and-lasso-regularisation-using-r/.

K. “A Gentle Introduction to Random Forests using R.” *Eight to Late*. September 20, 2016 (9:44 p.m.), https://eight2late.wordpress.com/2016/09/20/a-gentle-introduction-to-random- forests-using-r/.

“Lesson 3 Logistic Regression Diagnostics,” UCLA Institute for DIgital Research and Education. https://stats.idre.ucla.edu/stata/webbooks/logistic/chapter3/lesson-3-logistic-regression-diagnostics/.

LII Staff. “Workers Compensation.” LII / Legal Information Institute, Legal Information Institute, 4 Jan. 2018, www.law.cornell.edu/wex/workers_compensation.

Lunardon, N., Menardi, G., & Torelli, N. (2014, June).“ROSE: A package for Binary Imbalanced Learning.” *The R Journal*, 6/1.

“Notice of Options Following Disability Rating.” State of California Department of Industrial Relations Division of Workers” Compensation Disability Evaluation Unit. https:// www.dir.ca.gov/dwc/DEU110.pdf.

“Random Forests,” *Metagenomics. Statistics.*. https://dinsdalelab.sdsu.edu/metag.stats/code/randomforest.html.

Srivastava, Tavish. “7 Important Model Evaluation Error Metric Everyone should Know.” *Analytics Vidhya*. Last modified February 19, 2016. https://www.analyticsvidhya.com/blog/2016/02/7-important-model-evaluation-error-metrics/.

Appendicies

Appendix A - Data Adjustments

1. Kept the following Claim Types only: Temp Disability, Future Medical, Indemnity and Death.
2. Deleted observations that have program year 1994-1995 or prior. (no filtering/deleting needed for part 1 file)
3. Combined data file part 1 and part 2.
4. Deleted variables with only one single value.
5. Deleted Observations with NULL for "Claim Status at 6-30-2018".
6. Create a new variable named "Closure Year", which is the number of years between Date of Loss and Date Closed; Open claims are labeled as NA.
7. Average Weekly Wages: capped the wages at $10,000.
    a. Classified them into 5 intervals. (0, 1, 600, 900, 1250, 10000)
8. Age at DOL: Replaced ages that are less than 18 or greater than 75 years old by NA.
    a. Classified them into 4 intervals. (18, 37, 45, 53, 75)
9. Years Employed at DOL: Replaced negative, and unreasonable (e.g 111 years) values by NA, and change the value <1 to 0.5.
    a. Classified them into 4 intervals. (0.5, 2, 7, 14, 70)
10. Grouped variable into 4 groups with approximately equal sizes.
    a. Incurred Total. (0, 1000, 4000, 25000, 1160000)
11. Sorted data by frequency, set a cut-off point for the frequency, and any data below that point was grouped into one.
    a. Entity Type
    b. Class Code
    c. Nature of Injury Code
12. Body Part Code: grouped manually and logically (i.e. head/neck/back, arms, legs, trunk, multiple body parts, etc.), and create a mapping for referencing purpose.
13. Cause of Loss Code: grouped manually and logically (i.e. fall, burn), and create a mapping for referencing purpose.
14. Deleted: PD Rating, PD Incurred Flag, Future Medical, Settlement Amount, Paid Legal Flag, Paid Total, Reserve Total.
15. Created a "Reopened Check".
16. Deleted Occupation and used Class Code. Grouped Class Code by frequency.
17. Age at DOL Interval: Changed [18,37] to [18, 25] and [26,37]
18. Relabeled factor variables with NAs (created a new variable for each of them):
19. Grouped PD rating
    a. Grouped by logic: 0 in one group, all other values in another group, since majority of the ratings are 0s'.
    b. Dealt with miscodings (e.g put 181, 400 into the largest group)
20. Created "Pension Age" = Age at DOL + Years Employed
    a. Performed Grouping

i. Under age 50 in one group, 5 years interval after age 50, and over age 80 in another group.

21. Created “Age of Claim” = Evaluation Date - DOL = 6/30/2018 - Date of Loss
22. Created “Current Claimant’s Age” = Age of Claim + Age at DOL
    a. Performed Grouping (Same grouping technique as Age at DOL)

Appendix B - Data Profile

Variables Used

- Entity Type Group
- Gender
- Safety
- Claim Type
- Settlement Type
- Litigation
- Paid ALAE Over 5K Flag
- Reopened Check
- Class Code Label2
- Body Part Code Label2
- Cause of Loss Code Label2
- Closure Yr Interval Label2
- Weekly Wages Interval Label2
- Age at DOL Interval Label2
- Incurred Interval Label2
- Nature of Injury Code Label2
- Pension Age Interval Label2
- Years Employed at DOL Interval Label2

Variables Created

- Closure Year
- Generalized COL (generalized Cause of Loss Code)
- Area of Body (Grouped Body Part Code)
- Entity Type Group
- Closure Yr Interval
- Weekly Wages Interval
- Age at DOL interval
- Years Employed at DOL Interval
- Incurred Interval
- Pension Year
- PD Rating Interval
- Reopened Check
- Age of Claim
- Current Claimant’s Age
- Pension Year Interval
- Current Claimant’s Age Interval
- Class Code Label2
- Body Part Code Label2
- Cause of Loss Code Label2

- Closure Yr Interval Label2
- Weekly Wages Interval Label2
- Age at DOL Interval Label2
- Incurred Interval Label2
- Nature of Injury Code Label2
- Pension Age Interval Label2
- Years Employed at DOL Interval Label2

Variables Dropped

- Incurred Total
- Average Weekly Wages
- Entity Type
- Age at DOL
- Years Employed at DOL
- Occupation
- Client Number
- Client Name
- Submission ID
- Submission Name
- Entity ID
- Entity Name
- Entity Group
- Entity Sub-Type
- Entity Detail Type
- ProgramYear
- Location
- Department
- Master Claim Number
- Original Claim Number
- Date of Loss
- Date Reported
- Date Received
- Date Entered
- Cause of Loss Description
- Nature of Injury Description
- Body Part Description
- Loss Description
- Evaluation Date
- Age in Months
- Claim Status
- Claim Status at 6-30-2018
- Settlement Date
- Examiner Name
- PD Amount
- PD Paid Flag
- PD Incurred Flag
- PD First Check Date
- Future Medical Award
- Date Closed
- Class Code
- Cause of Loss Code
- Nature of Injury Code
- Body Part Code
- Settlement Amount
- PD Rating
- Fatality
- Paid Legal Flag
- Date Accepted
- Date Delayed
- Date Denied
- Days Paid 4850
- Days Paid 4850 Count
- Days Paid TD
- Days Paid TD Count
- Days Paid OSHA
- Days Paid OSHA Count
- Days Lost
- Days Lost Count
- Days Modified Duty
- Days Modified Duty Count
- Paid TD
- Paid PD
- Paid 4850
- Paid OtherIndemnity
- Paid Voc Rehab
- Paid Medical
- Paid 3 Year Medical
- Paid ALAE
- Paid Legal
- Paid Total
- Reserve TD
- Reserve PD
- Reserve 4850
- Reserve OtherIndemnity
- Reserve Voc Rehab
- Reserve Medical
- Reserve ALAE
- Reserve Legal
- Reserve Total

- Incurred TD
- Incurred PD
- Incurred 4850
- Incurred Other Indemnity
- Incurred Voc Rehab
- Incurred Medical
- Incurred ALAE
- Incurred Legal
- Recovery Subrogation
- Recovery Excess
- Recovery Total
- Open Claim Count
- Closed Claim Count
- Claim Count
- Incident/First Aid Claim Count
- Medical Only Claim Count
- Future Medical Claim Count
- Td Claim Count
- Pd Claim Count
- Ind Claim Count
- Open Incident/First Aid Claim Count
- Open Medical Only Claim Count
- Open Future Medical Claim Count
- Open TD Claim Count
- Open PD Claim Count
- Open Indemnity Claim Count
- Litigated Claim Count
- Accepted Claim Count
- Delayed Claim Count
- Denied Claim Count
- Settled Claim Count

Data Profile of Original Dataset

Data Columns

Column Headings	Type	Important?	Comments and Questions
Client Number	Numeric		25 categories in total, each of them is labeled as 4-digit number.
Client Name	Character		35 client names in total, "NCCSIF W/C" has most claims.
Submission ID	Numeric		376 categories in total, each of them is labeled as 3-digit number.
Submission Name	Character		32 submission names in total, each of them starting with "York".
Entity ID	Numeric		468 categories in total, with no more than 4 digits each.
Entity Name	Character		468 categories in total, corresponding to "Entity ID", which can be deleted later in prediction.
Entity Group	Character		Types of entities, containing 17 variables, education has most claims.
Entity Types	Character	Yes	9 variables, "Municipalities (Cities and Towns)" has most claims.
Entity Sub-types	Character		19 variables, similar to "Entity Group".
Entity Detail type	Character		51 variables, which are more specific.
Program Year	Year		From 1995-1996 to 2017-2018.
Location	Character		1806 variables, and contain lots of missing value.
Department	Character		2948 variables. Describe the exact working department.
Master Claim Number	Character		35615 variables. Consists of alpha-numeric characters which mark the claims.
Original Claim Number	Character		60015 variables.
Date of Loss	Date	Yes	
Date Reported	Date		
Date Received	Date		
Column Headings	**Type**	**Important?**	**Comments and Questions**
Date Entered	Date		
Gender	Character	Yes	Combine unknown and other. 9.76% of them are Unknown and other gender.

			Maybe the claim entailed more than one people or some other reasons.
Age at Dol	Numeric	Yes	96 age categories in total. Age -1 and 0 seem pretty confused.
Years Employed at DOL	Numeric	Maybe	years such as -1, -2, -3, -4 existed, which don't make sense.
Occupation	Character	Maybe	4146 variables existed. All of them are working titles.
Safety	Indicator	Maybe	2 variables 0 and 1. True indicates fire or police employees. 22.85% of claims are 1.
Class Code	Numeric	Maybe	Contains huge amount of missing values. Might not be a good predictor. (26%)
Average Weekly Wages	$	Yes	
Cause of Loss Code	Numeric	Yes	77 variables in total, including lots of missing value.
Cause of Loss Description	Text		148 variables in total. Should stay corresponded to Code but not.
Nature of Injury Code	Numeric	Yes	62 variables in total.
Nature of Injury Description	Text		115 variables in total. Same question as the Loss Description.
Body Part Code	Numeric	yes	58 variables.
Body Part Description	Text		115 variables.
Evaluation Date	Date		Annual evaluation on 06/30 from 1996 to 2018.
Age in Months	Numeric		Multiples of 12. 216 is the maximum.
Claim Status	Character		4 levels of data. This column is as of the evaluation date for the data record.
Claim Status at 6-30-2018	Character		4 levels of data. Evaluation date is on 6-30-2018.
Claim Type	Character	Yes	4 levels of data.
Settlement Type	Character	Yes	6 types of settlement.
Settlement Date	Date		Need a format transformation.
Column Headings	**Type**	**Important?**	**Comments and Questions**
Settlement Amount	$	Maybe	
Examiner Name	Name		330 variables in total.
PD Rating	Numeric	Maybe	100 levels. This is the estimated percentage of Permanent Disability for

			a claimant.
PD Amount	$	No?	Estimate of loss of permanent disability loss
PD Paid Flag	Indicator		Indicator to tell us if there was any PD paid on a claim
PD Incurred Flag	Indicator	Maybe	Flag for when we think the Incurred is PD
PD First Check Date	Date		Need a format transformation.
Future Medical Award	Indicator	Maybe	unpaid medical coming.
Date Closed	Date		Need a format transformation.
Fatality	Indicator		Almost all of the data are 0.
Litigation	Indicator	Yes	About 20% have litigation.
Paid Legal Flag	Indicator	Yes	About 10% are true. Don't know exactly what this means.
Paid ALAE Over 5K Flag	Indicator	Yes	About 10% are true. Don't know exactly what this means.
Date Accepted	Date		Need a format transformation.
Date Delayed	Date		Need a format transformation.
Date Denied	Date		Need a format transformation.
Days Paid 4850	Numeric		Don't know exactly what this means.
Days Paid 4850 Count	Indicator		About 5% of the data are true.
Days Paid TD	Numeric		
Days paid TD Count	Indicator		About 33.5% of the data are true.
Days Paid OSHA	Numeric		Don't know exactly what this means.
Days Paid OSHA Count	Indicator		About 41.5% of the data are true.
Days Modified Duty	Numeric		Don't know exactly what this means.
Days Modified Count	Indicator		About 7% of the data are true.
Paid TD	$		Paid Indemnity on a TD claim.
Paid PD	$		Paid Indemnity on a PD claim.
Paid 4850	$		Paid Indemnity on a 4850 claim.
Paid Voc Rehab	$		Don't know exactly what this means.
Paid Medical	$		
Column Headings	**Type**	**Important?**	**Comments and Questions**
Paid 3 Year Medical	$		Not needed for this project.
Paid ALAE	$		
Paid Legal	$		
Paid Total	$	yes	
Reserve TD	$		
Reserve PD	$		

Reserve 4850	$		
Reserve OtherIndemnity	$		
Reserve Rehab	$		
Reserve Medical	$		
Reserve ALAE	$		
Reserve Total	$	yes	
Incurred TD	$		
Incurred PD	$		
Incurred 4850	$		
Incurred OtherIndeminity	$		
Incurred Rehab	$		
Incurred Medical	$		
Incurred AlAE	$		
Incurred Legal	$		
Incurred Total	$	yes	
Recovery Subrogation	$		Medical Equipment; Liam Claim
Recovery Excess	$		Over SIR
Recovery total	$		
Open Claim Count	Indicator		
Closed Claim Count	Indicator		
Future Medical Claim Count	Indicator		
Td Claim Count	Indicator		
Open Future Medical Claim Count	Indicator		
Open TD Claim Count	Indicator		
Open Indemnity Claim Count	Indicator		
Litigated Claim Count	Indicator		
Accepted Claim Count	Indicator		
Column Headings	**Type**	**Important?**	**Comments and Questions**
Delayed Claim Count	Indicator		
Denied Claim Count	Indicator		
Settled Claim Count	Indicator		

Data Values

Program Year					
1995	-	1996		21950	5.51%
1996	-	1997		21153	5.31%
1997	-	1998		21545	5.41%
1998	-	1999		23118	5.80%
1999	-	2000		35781	8.98%
2000	-	2001		32375	8.13%
2001	-	2002		21994	5.52%
2002	-	2003		38107	9.57%
2003	-	2004		24417	6.13%
2004	-	2005		21871	5.49%
2005	-	2006		18799	4.72%
2006	-	2007		19500	4.90%
2007	-	2008		18272	4.59%
2008	-	2009		15229	3.82%
2009	-	2010		13548	3.40%
2010	-	2011		11434	2.87%
2011	-	2012		10522	2.64%
2012	-	2013		8387	2.11%
2013	-	2014		7294	1.83%
2014	-	2015		5255	1.32%
2015	-	2016		4042	1.01%
2016	-	2017		2526	0.63%
2017	-	2018		1205	0.30%
				398324	100.00%
Evaluation Date					
6/30/1999				4182	1.05%
6/30/2000				5945	1.49%
6/30/2001				7628	1.92%
6/30/2002				9037	2.27%
6/30/2003				11215	2.82%
6/30/2004				12930	3.25%
6/30/2005				14509	3.64%
Evaluation Date					
6/30/2006				15999	4.02%
6/30/2007				17621	4.42%

6/30/2008				19251	4.83%
6/30/2009				20776	5.22%
6/30/2010				22273	5.59%
6/30/2011				23666	5.94%
6/30/2012				25144	6.31%
6/30/2013				26471	6.65%
6/30/2014				29590	7.43%
6/30/2015				31070	7.80%
6/30/2016				31578	7.93%
6/30/2017				33979	8.53%
6/30/2018				35460	8.90%
				398324	100.00%
Claim Status					
Closed				274129	68.82%
Open				64931	16.30%
ReOpened-Closed				51982	13.05%
ReOpened-Open				7282	1.83%
				398324	100.00%
Claim Status at 6-30-2018					
Closed				304255	76.38%
Open				13822	3.47%
ReOpened-Closed				76079	19.10%
ReOpened-Open				4168	1.05%
				398324	100.00%
Claim Type					
Death				33	0.01%
Future Medical				57247	14.37%
Indemnity				48862	12.27%
Temp Disability				292182	73.35%
				398324	100.00%
Settlement Type					
Compromise & Release				20504	5.15%

Findings and Award				1094	0.27%
Not Settled				297046	74.57%
Other Settlement				18184	4.57%
Stipulated Award				35056	8.80%
Unknown / Other				26440	6.64%
				398324	100.00%

Data Profile of Modeling Dataset

Column Headings	**Number of levels**	**Comments**
Entity.Type.Group	4	
Gender	3	
Safety	2	Indicator, 0 and 1.
Claim.Type	4	
Settlement.Type	2	Target variable.
Litigation	2	Indicator, 0 and1.
Paid.ALAE.Over.5K.Flag	2	Indicator, 0 and1.
Reopened.Check	2	Indicator, 0 and1.
Class.Code.Label	7	
Body.Part.Code.Label	6	
Cause.of.Loss.Code.Label	9	
Closure.Yr.Interval.Label	5	
Weekly.Wages.Interval.Label	6	
Age.at.DOL.Interval.Label	6	
Incurred.Interval.Label	5	
Nature.of.Injury.Code.Label	9	
Pension.Age.Interval.Label	9	
Years.Employed.at.DOL.Interval.Label	5	

Definition of Labeling

Label Number	Label Name	Comment
Cause of Loss Code		
1	Misclassified	
2	Burn	
3	Caught	
4	Cut/Abrasion	
5	Fall	
6	Other	
7	Strain	
8	Strike	
9	Vehicle	
Class Code		
1	Misclassified	
2	7706	Fire
3	7720	Police
4	8875	Public colleges or schools
5	9410	Municipal/State/Public Agency Employees
6	9420	Municipal/State/Public Agency Others
7	Other	
Body Part Code		
1	Misclassified	
2	Arms	
3	Head/Neck/Back	
4	Legs	
5	Other/Multiple Injuries	
6	Trunk	
Nature of Injury Code		

1	Other	
2	10	Contusion
3	28	Fracture
4	49	Sprain
5	52	Strain
6	59	Hypertension
7	77	Mental Stress
8	80	All Other Cumulative Injuries
9	Misclassified	
Closure Year Interval		
1	Not in Interval	
2	[0,1)	
3	[1,2)	
4	[2,4)	
5	[4,43)	
Weekly Wages Interval		
1	Not in Interval	
2	[0,1)	
3	[1,600)	
4	[1,250, 10,000)	
5	[600,900)	
6	[900, 1,250)	
Age at DOL Interval		
1	Not in Interval	
2	[18,25)	
3	[25,37)	
4	[37,45)	
5	[35,53)	
6	[53,75)	
Years Employed at DOL Interval		

1	Not in Interval	
2	[0.5,2)	
3	[14,70)	
4	[2,7)	
5	[7,14)	
Incurred Interval		
1	Not in Interval	
2	(0, 1,000]	
3	(1000, 4,000]	
4	(25,000, 1,160,000]	
5	(4,000, 25,000]	
Pension Age Interval		
1	Not in Interval	
2	[0,50)	
3	[50,55)	
4	[55,60)	
5	[60,65)	
6	[65,70)	
7	[70,75)	
8	[75,80)	
9	>= 80	

Appendix C - Coding

```
#The coding below is used for the entire project. Further explanation is commented within the lines.
setwd("/Users/mingzhang/Desktop/pstat296")  # Set working directory
library(readxl)
tb1 = read_xlsx("Ucsb_06302018Part1.xlsx")   # read two datasets
tb2 = read_xlsx("Ucsb_06302018Part2.xlsx")
library(tidyverse)
library(dplyr)
tb1 = as.tibble(tb1)   #make them tibble for further implementing
tb2 = as.tibble(tb2)
tb2$`Date Received`=as.numeric(tb2$`Date Received`)    #change the type of date
tb2$`Date Received`=as.Date(tb2$`Date Received`,origin = "1899-12-30")
tb.new = rbind(tb1,tb2)  # combine two datasets
df1 = tb1
df2 = tb2
df1$rowID <- seq.int(nrow(df1))    # make row id for further tracing
df2$rowid <- seq.int(nrow(df2))
names(df2)[names(df2)=="rowid"] = "rowID"
df1 = df1[,c(118,1:117)]   # move rowID to the front
df2 = df2[,c(118,1:117)]
df1 = filter(df1, `Claim Type` %in% c("Temp Disability","Future Medical"))  #only keep 4 claim types
which have the possibility to become C&R
df2 = filter(df2, `Claim Type` %in% c("Temp Disability","Future Medical", "Indemnity", "Death"))
df2 = filter(df2, ProgramYear %in% c("1995-1996","1996-1997", "1997-1998", "1998-1999", "1999-
2000","2000-2001","2001-2002","2002-2003","2003-2004","2004-2005","2005-2006","2006-
2007","2007-2008","2008-2009","2009-2010","2010-2011","2011-2012","2012-2013","2013-
2014","2014-2015","2015-2016","2016-2017","2017-2018")) # Only keep recent program years data
df.new = rbind(df1,df2)
uns = vector()  # find variables which only have one level of value.
for(i in 1:118){
  ci = df.new[[i]]
  uns = c(uns, length(unique(ci)))
  }
uns
col_delete = NULL
for (i in 1:118) {
  if (uns[i] == 1) {
    col_delete = c(col_delete, i)
  }
}
col_delete
df.new = subset(df.new, select = -c(34,64,65,71,85,101,102,103,106,107,108,109,112)) # delete them
```

```
df.new = filter(df.new, `Claim Status at 6-30-2018` %in% c("Closed", "Open", "ReOpened-Closed", "ReOpened-Open")) # delete the null value in this column
df.new$`Settlement Date` = as.numeric(df.new$`Settlement Date`)   #change the date type
df.new$`PD First Check Date` = as.numeric(df.new$`PD First Check Date`)
df.new$`Date Closed` = as.numeric(df.new$`Date Closed`)
df.new$`Date Accepted` = as.numeric(df.new$`Date Accepted`)
df.new$`Date Delayed` = as.numeric(df.new$`Date Delayed`)
df.new$`Date Denied` = as.numeric(df.new$`Date Denied`)
df.new$`Settlement Date` = as.Date(df.new$`Settlement Date`,origin = "1899-12-30",format = "%Y.%m.%d")    #change the date type
df.new$`PD First Check Date` = as.Date(df.new$`PD First Check Date`, origin ="1899-12-30",format = "%Y.%m.%d" )
df.new$`Date Closed` = as.Date(df.new$`Date Closed`, origin ="1899-12-30",format = "%Y.%m.%d") #change the format of the date
df.new$`Date Accepted` = as.Date(df.new$`Date Accepted`, origin ="1899-12-30",format = "%Y.%m.%d")
df.new$`Date Delayed` = as.Date(df.new$`Date Delayed`, origin ="1899-12-30",format = "%Y.%m.%d")
df.new$`Date Denied`= as.Date(df.new$`Date Denied`, origin = "1899-12-30",format = "%Y.%m.%d")
df.new$`Date Received` = as.Date(df.new$`Date Received`, origin = "1899-12-30",format = "%Y.%m.%d")
df2018 = filter(df.new, `Evaluation Date` == as.Date(43281,origin ="1899-12-30")) #only keep the date at the recent evaluation date
df2018$`Evaluation Date` = as.Date(df2018$`Evaluation Date`, origin = "1899-12-30",format = "%Y.%m.%d")
setwd("/Users/mingzhang/Desktop/pstat296")
tb = read_xlsx("updated_file20181128.xlsx")
df = tb
`Reopened Check` = ifelse(df$`Claim Status` == "ReOpened-Closed"|df$`Claim Status` =="ReOpened-Open", 1, 0)  # set a reopened check
table(`Reopened Check`)
df$`Reopened Check` = `Reopened Check`
df = filter(df, `Evaluation Date` == as.Date(43281,origin ="1899-12-30"))
df[df=="NA"] = NA
df$`Evaluation Date` = as.Date(df$`Evaluation Date`)
df$`Settlement Date` = as.Date(df$`Settlement Date`)
df$`PD First Check Date` = as.Date(df$`PD First Check Date`)
df$`Date Closed` = as.Date(df$`Date Closed`)
df$`Date Accepted` = as.Date(df$`Date Accepted`)
df$`Date Delayed` = as.Date(df$`Date Delayed`)
df$`Date Denied`= as.Date(df$`Date Denied`)
df$`Date Received` = as.Date(df$`Date Received`)
df$`Date of Loss` = as.Date(df$`Date of Loss`)
df$`Date Reported` = as.Date(df$`Date Reported`)
df$`Date Entered` = as.Date(df$`Date Entered`)
```

```
date_diff = df$`Settlement Date` - df$`Date of Loss`    # calculate the difference between these two dates
year_diff = round(date_diff/365)   # make the year difference integer
df$`Closure Year` = year_diff   # change the column
df.new = subset(df, select = c("Entity Type", "Closure Year", "Gender", "Average Weekly Wages", "Age at DOL", "Years Employed at DOL", "Occupation", "Safety", "Class Code", "Cause of Loss Code", "Nature of Injury Code", "Body Part Code","Claim Type", "Settlement Type", "Settlement Amount", "PD Rating", "PD Incurred Flag", "Future Medical Award", "Litigation", "Paid Legal Flag", "Paid ALAE Over 5K Flag", "Paid Total", "Incurred Total", "Reserve Total", "Reopened Check"))   # Pick the useful columns
new_entity_type = df.new$`Entity Type`
new_entity_type = as.factor(new_entity_type)  #factorize it
#levels(new_entity_type)
new.levels = c("Other", "Other", "Municipalities (Cities and Towns)", "Other",
               "Other", "Other", "Other", "School Districts", "Special Districts")    #levels changing
`New Entity Type` = factor(new.levels[new_entity_type])

df.new = add_column(df.new, `New Entity Type`, .after = "Entity Type")  #add new column
df.new$`Class Code`[which(df.new$`Class Code` =="NULL")] = NA  # change NULL into na

df.new$`Class Code`[which(df.new$`Class Code` !="9410" & df.new$`Class Code` !="9420" & df.new$`Class Code` !="7706" & df.new$`Class Code` !="7720" & df.new$`Class Code` !="8875")] = "Other"    #pick several highest frequency levels

df.new$`Nature of Injury Code`[which(df.new$`Nature of Injury Code` != "52" & df.new$`Nature of Injury Code` != "49" & df.new$`Nature of Injury Code` != "10" & df.new$`Nature of Injury Code` != "80" & df.new$`Nature of Injury Code` != "77" & df.new$`Nature of Injury Code` != "28" & df.new$`Nature of Injury Code` != "59")] = "Other"   #pick several highest frequency levels
df.new$`Closure Year` = as.numeric(df.new$`Closure Year`)

`Closure Yr Interval` = cut(df.new$`Closure Year`, c(0,1,2,4,43), right = FALSE) # cut this column into intervals

df.new = add_column(df.new, `Closure Yr Interval`, .after = "Closure Year")  # add new column

colnames(df.new)[2] = "Entity Type Group"   # rename
colnames(df.new)[9] = "Age at DOL interval"
colnames(df.new)[11] = "Years Employed at DOL Interval"

df.new$`Cause of Loss Code` = as.character(df.new$`Cause of Loss Code`)   #classify cause of loss code according to each meaning
df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("1", "2", "3", "4", "5", "6", "7", "8", "9", "11", "84"))] = "Burn"
df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("12", "14"))] = "Burn"
```

```
df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("10", "13", "20"))] =
"Caught"

df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("15","16", "17", "18", "19",
"94", "95"))] = "Cut/Abrasion"

df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("25", "26", "27", "28", "29",
"30", "31", "32", "33"))] = "Fall"

df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("40", "41", "45", "46", "48",
"50"))] = "Vehicle"

df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("52", "53", "54", "55", "56",
"57", "58", "59", "60", "61", "97"))] = "Strain"

df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("65", "66", "67", "68",
"70"))] = "Strike"

df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("69", "74", "75", "76", "77",
"78", "79", "80", "81", "82", "85", "86", "87", "88", "89", "90", "92", "96", "98", "99"))] = "Other"
df.new$`Cause of Loss Code`[which(df.new$`Cause of Loss Code` %in% c("0"))] = "Other"

table(df.new$`Cause of Loss Code`)

tb_data = read.csv("updated_file0204.csv")
tb_data_export = NULL
tb_data_export = tb_data[,-which(names(tb_data) %in% na_columns)]   # find na columns
tb_data_export2 = tb_data[,-which(names(tb_data) %in% na_columns)]
na_columns <- colnames(tb_data)[colSums(is.na(tb_data)) > 0]
write.csv(tb_data3, file = "updated_file0212_2.csv", row.names = FALSE)

colnames(tb_data)
na_columns <- colnames(tb_data)[colSums(is.na(tb_data)) > 0]

#NA is labeled as 99 for numerical data, "Not in Interval" for interval grouping
tb_data$Closure.Year.Label <- tb_data$Closure.Year
tb_data$Closure.Year.Label[is.na(tb_data$Closure.Year.Label)] <- 99

tb_data$Closure.Yr.Interval.Label <-  addNA(tb_data$Closure.Yr.Interval)
levels(tb_data$Closure.Yr.Interval.Label) <-  c(levels(tb_data$Closure.Yr.Interval), "Not in Interval")

tb_data$Weekly.Wages.Interval.Label <-  addNA(tb_data$Weekly.Wages.Interval)
levels(tb_data$Weekly.Wages.Interval.Label) <-  c(levels(tb_data$Weekly.Wages.Interval), "Not in
Interval")
```

```
tb_data$Age.at.DOL.Label <- tb_data$Age.at.DOL
tb_data$Age.at.DOL.Label[is.na(tb_data$Age.at.DOL.Label)] <- 99

tb_data$Age.at.DOL.Interval.Label <- addNA(tb_data$Age.at.DOL.interval)
levels(tb_data$Age.at.DOL.Interval.Label) <-  c(levels(tb_data$Age.at.DOL.interval), "Not in Interval")

tb_data$Years.Employed.at.DOL.Label <- tb_data$Years.Employed.at.DOL
tb_data$Years.Employed.at.DOL.Label[is.na(tb_data$Years.Employed.at.DOL.Label)] <- 99

tb_data$Years.Employed.at.DOL.Interval.Label <- addNA(tb_data$Years.Employed.at.DOL.Interval)
levels(tb_data$Years.Employed.at.DOL.Interval.Label) <-
c(levels(tb_data$Years.Employed.at.DOL.Interval), "Not in Interval")

tb_data$Class.Code.Label <- addNA(tb_data$Class.Code)
levels(tb_data$Class.Code.Label) <-  c(levels(tb_data$Class.Code), "Misclassified")

tb_data$Cause.of.Loss.Code.Label <- addNA(tb_data$Cause.of.Loss.Code)
levels(tb_data$Cause.of.Loss.Code.Label) <-  c(levels(tb_data$Cause.of.Loss.Code), "Misclassified")

tb_data$Nature.of.Injury.Code.Label <- addNA(tb_data$Nature.of.Injury.Code)
levels(tb_data$Nature.of.Injury.Code.Label) <-  c(levels(tb_data$Nature.of.Injury.Code), "Misclassified")

tb_data$Body.Part.Code.Label <- addNA(tb_data$Body.Part.Code)
levels(tb_data$Body.Part.Code.Label) <-  c(levels(tb_data$Body.Part.Code), "Misclassified")

tb_data$Incurred.Interval.Label <- addNA(tb_data$Incurred.Interval)
levels(tb_data$Incurred.Interval.Label) <-  c(levels(tb_data$Incurred.Interval), "Not in Interval")

tb_data$Pension.Age = as.double(tb_data$Age.at.DOL) + tb_data$Years.Employed.at.DOL
table(tb_data$Pension.Age)
table(tb_data$PD.Rating)

tb_data$Pension.Age.Interval <- cut(tb_data$Pension.Age, breaks = c(18,50,55,60,65,70,75,80,140))
tb_data$Pension.Age.Interval.Label <- addNA(tb_data$Pension.Age.Interval)
levels(tb_data$Pension.Age.Interval.Label) <-  c(levels(tb_data$Pension.Age.Interval), "Not in Interval")

#NA in Pension Age is labeled as 999
tb_data$Pension.Age.Label <- tb_data$Pension.Age
tb_data$Pension.Age.Label[is.na(tb_data$Pension.Age.Label)] <- as.double(999)
#PD Rating is labeled as either 0, or Other
tb_data$PD.Rating.Interval <- cut(tb_data$PD.Rating, breaks = c(0,1,400), labels = c("0", "Other"),
include.lowest = TRUE, right = FALSE)
```

```
#Unreasonable values are grouped to the largest group
tb_data$PD.Rating.Interval[tb_data$PD.Rating == 181] <- "0"
tb_data$PD.Rating.Interval[tb_data$PD.Rating == 400] <- "0"
tb_data_import2 = read.csv("updated_file0206.csv")
colnames(tb_data_import2)[colSums(is.na(tb_data_import2)) > 0]
#Levels are labeled as letters; level "A" is always NA
tb_data2 <- select(tb_data_import2, -c(Entity.Type, Occupation, PD.Rating, Date.of.Loss,
Closure.Year.Label, Age.at.DOL.Label, Years.Employed.at.DOL.Label, Pension.Age.Label))

colnames(tb_data2)

tb_data2$Class.Code.Label2 <- tb_data2$Class.Code.Label
levels(tb_data2$Class.Code.Label2) <- c("B", "C", "D", "E", "F", "G", "A")
levels(tb_data2$Class.Code.Label2)

tb_data2$Body.Part.Code.Label2 <- tb_data2$Body.Part.Code.Label
levels(tb_data2$Body.Part.Code.Label2) <- c("B", "C", "D", "A", "E", "F")

tb_data2$Cause.of.Loss.Code.Label2 <- tb_data2$Cause.of.Loss.Code.Label
levels(tb_data2$Cause.of.Loss.Code.Label2) <- c("B", "C", "D", "E", "A", "F", "G", "H", "I")

tb_data2$Closure.Yr.Interval.Label2 <- tb_data2$Closure.Yr.Interval.Label
levels(tb_data2$Closure.Yr.Interval.Label2) <- c("B", "C", "D", "E", "A")

tb_data2$Weekly.Wages.Interval.Label2 <- tb_data2$Weekly.Wages.Interval.Label
levels(tb_data2$Weekly.Wages.Interval.Label2) <- c("B", "C", "D","E", "F", "A")

tb_data2$Age.at.DOL.Interval.Label2 <- tb_data2$Age.at.DOL.Interval.Label
levels(tb_data2$Age.at.DOL.Interval.Label2) <- c("B", "C", "D", "E", "F", "A")

tb_data2$Incurred.Interval.Label2 <- tb_data2$Incurred.Interval.Label
levels(tb_data2$Incurred.Interval.Label2)
levels(tb_data2$Incurred.Interval.Label2) <- c("B", "C", "D", "E", "A")

tb_data2$Nature.of.Injury.Code.Label2 <- tb_data2$Nature.of.Injury.Code.Label
levels(tb_data2$Nature.of.Injury.Code.Label2)
levels(tb_data2$Nature.of.Injury.Code.Label2) <- c("B", "C", "D", "E", "F", "G", "H", "A", "I")

tb_data2$Pension.Age.Interval.Label2 <- tb_data2$Pension.Age.Interval.Label
levels(tb_data2$Pension.Age.Interval.Label2)
levels(tb_data2$Pension.Age.Interval.Label2) <- c("B", "C","D","E", "F","G", "H","I","A")

tb_data2$Years.Employed.at.DOL.Interval.Label2 <- tb_data2$Years.Employed.at.DOL.Interval.Label
levels(tb_data2$Years.Employed.at.DOL.Interval.Label2)
```

```
levels(tb_data2$Years.Employed.at.DOL.Interval.Label2) <- c("B", "C", "D", "E", "A")

#tb_data3 excludes columns with the original labeling
tb_data3 <- select(tb_data2, -c(Closure.Yr.Interval.Label, Weekly.Wages.Interval.Label,
Age.at.DOL.Interval.Label, Years.Employed.at.DOL.Interval.Label, Class.Code.Label,
Cause.of.Loss.Code.Label, Nature.of.Injury.Code.Label, Body.Part.Code.Label, Incurred.Interval.Label,
Pension.Age.Interval.Label))

#levels are relabeled as numbers; 1 is always NA
levels(tb_data3$Class.Code.Label2) <- c(2:7,1)
levels(tb_data3$Class.Code.Label2)

levels(tb_data3$Body.Part.Code.Label2) <- c(2:4, 1, 5,6)

levels(tb_data3$Cause.of.Loss.Code.Label2) <- c(2:5, 1, 6:9)

levels(tb_data3$Closure.Yr.Interval.Label2) <- c(2:5,1)

levels(tb_data3$Weekly.Wages.Interval.Label2) <- c(2:6, 1)

levels(tb_data3$Age.at.DOL.Interval.Label2) <- c(2:6,1)

levels(tb_data2$Incurred.Interval.Label2)
levels(tb_data3$Incurred.Interval.Label2) <- c(2:5,1)

levels(tb_data2$Nature.of.Injury.Code.Label2)
levels(tb_data3$Nature.of.Injury.Code.Label2) <- c(2:8,1, 9)

levels(tb_data2$Pension.Age.Interval.Label2)
levels(tb_data3$Pension.Age.Interval.Label2) <- c(2:9,1)

levels(tb_data2$Years.Employed.at.DOL.Interval.Label2)
levels(tb_data3$Years.Employed.at.DOL.Interval.Label2) <- c(2:5,1)
#Settlement type 0 = All Other, 1 = C&R
table(tb_data3$Settlement.Type)

levels(tb_data3$Settlement.Type) = list("1" = "Compromise & Release", "0" = c("Findings and Award",
"Not Settled", "Other Settlement", "Stipulated Award"))    # make the target variable into 0 and 1

names(tb_data3)
tb_data3 = subset(tb_data3, select = -c(3, 8))
head(tb_data3)
table(tb_data3$Litigation)
```

```
#Convert the data type from numeric to factor
cols_factor = c("Safety", "Paid.ALAE.Over.5K.Flag","Litigation", "Reopened.Check")
tb_data3[cols_factor] = lapply(tb_data3[cols_factor], as.factor)

table(tb_data3$Litigation)

library(randomForest)    # modeling library
library(rpart)
library(gbm)
library(ISLR)
library(dplyr)
library(tree)
library(maptree)
library(MASS)
library(e1071)
library(glmnet)
library(ROCR)
library(caret)
#load('df.new.Rda')

df.final = read.csv("updated_file0212_2.csv")
df.model = df.final
colnames(df.model)[ apply(df.model, 2, anyNA)]    #check if there is still na left
#df.model$Settlement.Type = as.character(df.model$Settlement.Type)
df.model = subset(df.model, select= -Incurred.Total)    #delete three columns
df.model = subset(df.model, select= -Average.Weekly.Wages)
df.model = subset(df.model, select= -PD.Rating.Interval)

table(df.model$Settlement.Type)
#cols_factor = c("Safety","Paid.ALAE.Over.5K.Flag", "Reopened.Check", "Litigation",
"Settlement.Type", "Class.Code.Label2", "Body.Part.Code.Label2",
"Cause.of.Loss.Code.Label2","Incurred.Interval.Label2", "Nature.of.Injury.Code.Label2" )
cols_factor = names(df.model)
df.model[cols_factor] = lapply(df.model[cols_factor], as.factor)   #factorize all columns
#df.model$Settlement.Type = as.numeric(df.model$Settlement.Type)
head(df.model)
df.model$Claim.Type = factor(df.model$Claim.Type,levels(df.model$Claim.Type)[c(4,1,2,3)]) #change
the level order of claim type
levels(df.model$Claim.Type)
library(ROSE)
set.seed(1)
n = nrow(df.model)
#names(df.model) = gsub(" ", "_", names(df.model))
```

```
in.trn= sample.int(n, 0.8*n)
df.train = df.model[ in.trn,]    # pick training and testing dataset
df.test = df.model[-in.trn,]

data.rose <- ROSE(Settlement.Type ~ ., data = df.train)$data   #balanced the training dataset
table(data.rose$Settlement.Type)
#data.ovun <- ovun.sample(Settlement_Type ~ ., N= 16000, p = 0.5, data = df.train)$data

#data.rose

cols_numeric = c("Closure.Yr.Interval.Label","Weekly.Wages.Interval.Label",
"Age.at.DOL.Interval.Label", "Incurred.Interval.Label", "Pension.Age.Interval.Label",
"Years.Employed.at.DOL.Interval.Label")    #change some columns into numeric ones for better
computing
data.rose[cols_numeric] = lapply(data.rose[cols_numeric], as.numeric)
df.test[cols_numeric] = lapply(df.test[cols_numeric], as.numeric)

set.seed(1)
fit.tree = tree(Settlement.Type~., data = data.rose, method = "class")   #building tree model
# control = tree.control(2456,minsize = 5, mindev = 1e-5)
calc_error_rate = function(predicted.value, true.value){    # writing a function calculating error rate
  return(mean(true.value!=predicted.value))
}

draw.tree(fit.tree, nodeinfo=TRUE,cex = 0.5 )   #plot the tree

Xtrain = subset(data.rose, select = -c(Settlement.Type))   # separate the x and y
Ytrain = data.rose$Settlement.Type
Xtest = subset(df.test, select = -c(Settlement.Type))
Ytest = df.test$Settlement.Type

pred1 = predict(fit.tree, Xtest, type="vector")       # get prediction probability
pred1.train = predict(fit.tree, Xtrain, type="vector")

pred.tree.train =factor(ifelse(pred1.train[,2]< 0.5, 0, 1))    #set the threshold and get binary response
pred.tree.test = factor(ifelse(pred1[,2]< 0.5, 0, 1))

tree.train.error = calc_error_rate(pred.tree.train, Ytrain)   # calculate error rate
tree.test.error = calc_error_rate(pred.tree.test, Ytest)

confusionMatrix(pred.tree.test, Ytest, positive = "1")  #building confusion matrix

set.seed(1)
fit.glm = glm(Settlement.Type ~., data = data.rose, family = "binomial")  # building logistic regression
```

```
prob.train.glm = predict(fit.glm, Xtrain , type="response")
train.result.glm = factor(ifelse(prob.train.glm < 0.53, 0, 1))
#calc_error_rate(train.result.glm, Ytrain)

prob.test.glm = predict(fit.glm, Xtest, type = "response")
test.result.glm = factor(ifelse(prob.test.glm < 0.53, 0, 1))

glm.train.error = calc_error_rate(train.result.glm, Ytrain)
glm.test.error = calc_error_rate(test.result.glm, Ytest)



summary(fit.glm)   #summary the fit and observe the slope and variable importance
confusionMatrix(test.result.glm, Ytest,positive = "1")

set.seed(1)

fit.bag = randomForest(Settlement.Type ~., data=data.rose, mtry=ncol(df.train)-1, importance=TRUE)
#build bagging model

pred2 = predict(fit.bag, Xtest, type = "prob")
pred2.train = predict(fit.bag, Xtrain, type = "prob")

train.result.bag = factor(ifelse(pred2.train[,2] < 0.22, 0, 1))
test.result.bag = factor(ifelse(pred2[,2] < 0.22, 0, 1))

bag.train.error = calc_error_rate(train.result.bag, Ytrain)
bag.test.error = calc_error_rate(test.result.bag, Ytest)

confusionMatrix(test.result.bag, Ytest,positive = "1")

set.seed(1)

fit.rf = randomForest(Settlement.Type ~., data=data.rose, mtry=6, importance=TRUE)
#build random forest
pred3 = predict(fit.rf, Xtest, type = 'prob')
pred3.train = predict(fit.rf, Xtrain, type = 'prob')

train.result.rf = factor(ifelse(pred3.train[,2] < 0.21, 0, 1))
test.result.rf = factor(ifelse(pred3[,2] < 0.21, 0, 1))

rf.train.error = calc_error_rate(train.result.rf, Ytrain)
rf.test.error = calc_error_rate(test.result.rf, Ytest)
```

```
varImpPlot(fit.rf,type = 2, sort=T, main="Variable Importance for random forest",cex = 0.8, n.var=10)   # make the variable importance plot
confusionMatrix(test.result.rf, Ytest,positive = "1")

set.seed(1)
fit.boost = gbm(ifelse(Settlement.Type == "1", 1, 0) ~., data=data.rose, distribution="bernoulli", n.trees=500, cv.folds = 5, interaction.depth = 4)
#build boosting model
summary(fit.boost)
train.prob.boost = predict.gbm(fit.boost, newdata = Xtrain, n.trees = 500, type = "response")
test.prob.boost = predict.gbm(fit.boost, newdata = Xtest, n.trees = 500,type = "response")

train.result.boost = factor(ifelse(train.prob.boost < 0.57262, 0, 1))
test.result.boost = factor(ifelse(test.prob.boost < 0.57262, 0, 1))

boost.train.error = calc_error_rate(train.result.boost, Ytrain)
boost.test.error =  calc_error_rate(test.result.boost, Ytest)

confusionMatrix(test.result.boost, Ytest, positive = "1")

set.seed(1)
#svm.tune = tune(svm, Settlement_Type~., data=df.train, kernel="linear",
#              ranges=list(cost=c(0.1,10,1000)))
#best.parameter = summary(tune.out)$"best.parameters"
fit.svm = svm(Settlement.Type~., data=data.rose, kernel="linear", cost=0.1,scale=FALSE)


fit.svm.prob = svm(Settlement.Type~., data=data.rose, kernel="linear", cost=0.1,probability = TRUE,scale=FALSE)
#build support vector machine model
pred4 = predict(fit.svm.prob, Xtest, probability = TRUE)
pred4.train = predict(fit.svm.prob, Xtrain, probability = TRUE)

train.result.svm = factor(ifelse(attr(pred4.train, which ="probabilities")[,2] < 0.5, 0, 1))
test.result.svm = factor(ifelse(attr(pred4, which ="probabilities")[,2] < 0.5, 0, 1))

svm.train.error = calc_error_rate(train.result.svm, Ytrain)
svm.test.error =  calc_error_rate(test.result.svm, Ytest)

confusionMatrix(test.result.svm, Ytest, positive = "1")

set.seed(1)
```

```
cv.out.ridge=cv.glmnet(model.matrix(~.,Xtrain), Ytrain, alpha = 1, family = "binomial")
bestlam = cv.out.ridge$lambda.min    # use cross validation to find the best lambda
fit.lasso = glmnet(model.matrix(~.,Xtrain), Ytrain, alpha = 1, lambda = bestlam, family = "binomial")
#build lasso regression model
train.pred.lasso = predict(fit.lasso, s=bestlam, newx = model.matrix(~.,Xtrain), type = "response")
train.result.lasso = factor(ifelse(train.pred.lasso<0.52, 0,1))

test.pred.lasso = predict(fit.lasso, s=bestlam, newx = model.matrix(~.,Xtest), type = "response")
test.result.lasso = factor(ifelse(test.pred.lasso<0.52, 0,1))

lasso.train.error = calc_error_rate(train.result.lasso, Ytrain)
lasso.test.error = calc_error_rate(test.result.lasso, Ytest)

confusionMatrix(test.result.lasso, Ytest, positive = "1")
set.seed(1)
cv.out.EN=cv.glmnet(model.matrix(~.,Xtrain), Ytrain, alpha = 0.5, family = "binomial")
bestlam1 = cv.out.EN$lambda.min
fit.EN = glmnet(model.matrix(~.,Xtrain), Ytrain, alpha = 0.5, lambda = bestlam1, family = "binomial")
#build elastic net model
train.pred.EN = predict(fit.EN, s=bestlam, newx = model.matrix(~.,Xtrain), type = "response")
train.result.EN = factor(ifelse(train.pred.EN<0.52, 0,1))

test.pred.EN = predict(fit.EN, s=bestlam, newx = model.matrix(~.,Xtest), type = "response")
test.result.EN = factor(ifelse(test.pred.EN<0.52, 0,1))

EN.train.error = calc_error_rate(train.result.EN, Ytrain)
EN.test.error = calc_error_rate(test.result.EN, Ytest)

confusionMatrix(test.result.EN, Ytest, positive = "1")

tree.pred = prediction(pred1[,2], Ytest)  #standardize prediction value
perf.tree = performance(tree.pred, measure = 'tpr', x.measure = 'fpr')  #use performance to get roc
glm.pred = prediction(prob.test.glm, Ytest)
perf.glm = performance(glm.pred, measure = 'tpr', x.measure = 'fpr')
bag.pred = prediction(pred2[,2], Ytest)
perf.bag = performance(bag.pred, measure = "tpr", x.measure = 'fpr')
rf.pred = prediction(pred3[,2], Ytest)
perf.rf = performance(rf.pred, measure = 'tpr', x.measure = 'fpr')
boosting.pred = prediction(test.prob.boost, Ytest)
perf.boosting = performance(boosting.pred,measure = 'tpr', x.measure = 'fpr')
svm.pred = prediction(attr(pred4, which ="probabilities")[,2], Ytest)
perf.svm = performance(svm.pred, measure = 'tpr', x.measure = 'fpr')
lasso.pred = prediction(test.pred.lasso, Ytest)
perf.lasso = performance(lasso.pred, measure = 'tpr', x.measure = 'fpr')
```

```
EN.pred = prediction(test.pred.EN, Ytest)
perf.EN = performance(EN.pred, measure = 'tpr', x.measure = 'fpr')

plot(perf.tree, col = "red", main="ROC")  #plot roc curve
plot(perf.glm, add = TRUE, col="blue")
plot(perf.bag, add = TRUE, col = "purple")
plot(perf.rf, add = TRUE, col = "black")
plot(perf.boosting, add = TRUE, col = "orange")
plot(perf.svm, add = TRUE, col = "pink")
plot(perf.lasso, add = TRUE, col = "yellow")
plot(perf.EN, add = TRUE, col = "grey")
legend(0.7,0.8, legend=c("decision tree", "logistic","bagging", "random forest", "boosting", "SVM",
"lasso", "elastic net"),
      col=c("red", "blue","purple", "black", "orange", "pink", "yellow", "grey"), lty=1:2, cex=0.8)

auc.tree = unlist(slot(performance(tree.pred, measure ='auc'),"y.values")) #get auc values
auc.glm = unlist(slot(performance(glm.pred, measure = 'auc'), "y.values"))
auc.bag = unlist(slot(performance(bag.pred, measure = 'auc'), "y.values"))
auc.rf = unlist(slot(performance(rf.pred, measure = 'auc'), "y.values"))
auc.boosting = unlist(slot(performance(boosting.pred, measure = 'auc'), "y.values"))
auc.svm = unlist(slot(performance(svm.pred, measure = 'auc'), "y.values"))
auc.lasso = unlist(slot(performance(lasso.pred, measure = 'auc'), "y.values"))
auc.EN = unlist(slot(performance(EN.pred, measure = 'auc'), "y.values"))

train.error = c(tree.train.error, glm.train.error, bag.train.error, rf.train.error, boost.train.error,
svm.train.error, lasso.train.error, EN.train.error)   #build error matrix
test.error = c(tree.test.error, glm.test.error, bag.test.error,rf.test.error, boost.test.error, svm.test.error,
lasso.test.error, EN.test.error)
error.matrix = cbind(train.error, test.error)
rownames(error.matrix) = c("decision tree", "logistic regression", "bagging", "random forest", "boosting",
"SVM", "lasso regression", "elastic net")
error.matrix

auc.list = c(auc.tree, auc.glm, auc.bag, auc.rf, auc.boosting, auc.svm, auc.lasso,auc.EN)
auc.matrix =as.matrix(auc.list)
rownames(auc.matrix) = c("decision tree", "logistic regression", "bagging", "random forest", "boosting",
"SVM", "lasso regression", "elastic net")
colnames(auc.matrix) = "auc.value"
auc.matrix
```