DATA MINING AND MACHINE LEARNING ALGORITHMS FOR WORKERS' COMPENSATION EARLY SEVERITY PREDICTION

---

by

David George Mathews

August 2016

A Thesis

Presented to the Faculty of the Department of Mathematical Sciences

Middle Tennessee State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Mathematical Sciences

Thesis Committee:

Don Hong, Chair

Qiang Wu

Lisa Green

James Hart

This thesis is dedicated to my wife Jincy, whose encouragement, support, and understanding made this research possible.

ACKNOWLEDGMENTS

My heartfelt gratitude to Dr. Tim Coomer, Al Rhodes ACAS, Michelle Bradley ACAS, and Dr. Lu Xiong at SIGMA Actuarial Consulting Group, Inc. for their support and understanding during this research project. Moreover, I express my sincere appreciation to Dr. Don Hong and Dr. Lisa Green for their valuable inputs and guidance. I express my gratitude to Dr. Qiang Wu ASA for his continued support throughout this research and for a groundbreaking recommendation that significantly improved predictive accuracy. I am also indebted to Peter McClellan, J.D., for proofreading this thesis.

ABSTRACT

Although the number of workers' compensation claims have been declining over the last two decades, average cost per claim has been steadily increasing. Identifying factors that contribute to severe claims and effectively managing those claims early in the claim life-cycle could reduce cost for employers and insurers. This research project utilizes machine learning algorithms to predict a binary severity outcome variable. A text mining algorithm, Correlated Topics Model, was used to convert textual description fields to topics. Support Vector Machines and Regularized Logistic Regression were implemented for severity classification and variable selection, respectively. Due to asymmetric severity outcomes in the training data, a balancing method for matching the volume of severe/non-severe claims was employed. Optimal model parameters for both algorithms were selected based on a profitability metric and 10-fold cross-validation. Discussion of data processing techniques and mathematical exposition of machine learning algorithms are provided. Open source statistical programming software, R, was utilized in this project.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

In the year 2013 workplace injury cost employers $88.5 billion($1.37 per $100 of covered wages) [5]. Among the three classes of workers compensation (WC) cases – medical only, temporary disability, and permanent disability – permanent disability cases, although rare, resulted in over 58% of cash benefits paid [5]. Sources of WC insurance include private insurance, state funds, or self-insurance. Although the number of work related claims have been declining over the last two decades, average cost per claim has been steadily increasing [7]. According to NCCI's State of the Line report, combined ratio of WC insurers and state funds stood at 98%, and 115%, respectively, in 2014 [6]; see Figure 1. Early intervention measures taken on claims which have the propensity to become severe could reduce cost for employers and insurers. This research project attempts to identify and investigate some commonly accepted predictors that lead to escalation of claim severity. Effective data processing techniques and machine learning algorithms are also discussed. A brief mathematical exposition of algorithms utilized is provided.

1.1 Predictors

1.1.1 Commonly Accepted Predictors

Thirty predictors – Opioids usage, age, depression, obesity, diabetes, addiction, smoking, litigation, lag between injury and filing claim, previous injury, stability of living arrangements, early hospitalization, number of doctor visits, prescription drug usage, occupation, specific doctor seen, therapies, injury type, hours worked, job title, commuting distance to job, emergency room visit, tenure, marital status, injury timing, pre-existing conditions, and treatment patterns – were mentioned in a majority of scholarly literature on the topic [7, 9, 10].

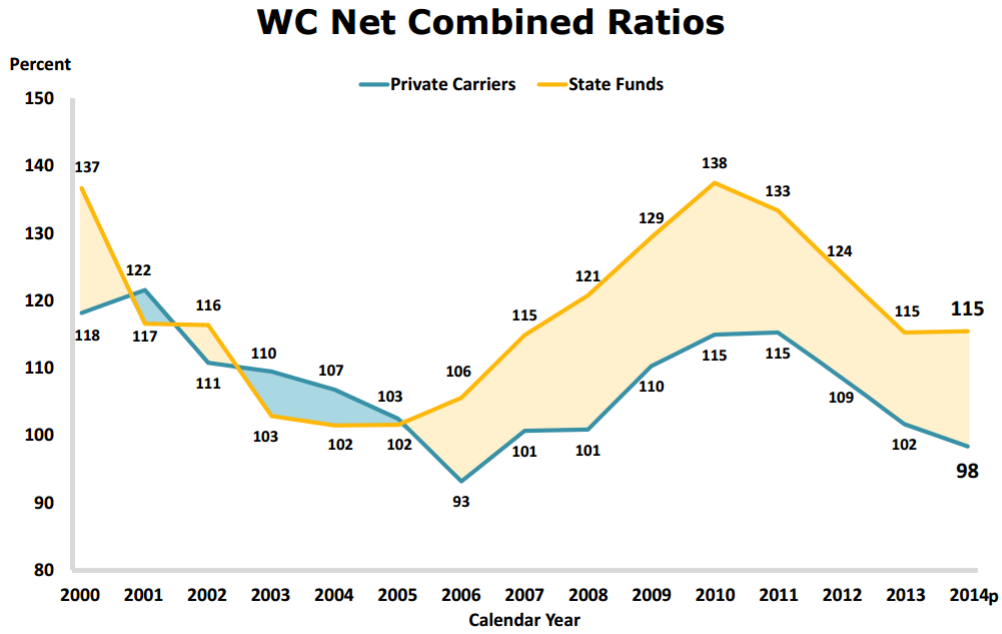

Figure 1: *NCCI State of Workers' Compensation Line 2015*

1.1.2 Predictors in Research Data

This study used simulated data based on observing real loss run data of self insured entities. The data set contains 27 predictors of both numeric and categorical types, and a dichotomous outcome variable indicating severity of claim. While some variables were directly used, others had to be calculated from the raw data. Gender, line of business, full time status, and geographic location were used directly. Other predictors, namely, age, tenure, and lag in reporting were calculated from date fields. Some predictors related to the injury were the object involved, bodily reaction or effect, and body part affected. Data fields containing description of events or injuries were assigned topic numbers using a topic modeling algorithm. Binary indicators were assigned for attorney involvement or litigated claims.

1.2 Data Inspection & Preparation

1.2.1 Type Conversion

The data file is converted from .xlsx to .csv format. Loss run data is inspected 10 predictors at a time, and employee names, serial numbers, and other unique fields are removed. Dates are formatted to become compatible with R. R compatible dates are required to allow arithmetic operations, such as subtraction on dates. Categorical predictors which are numbered and not ordinal are concatenated with a character to enable R to recognize the categorical nature of data. Fields having dollar signs are not recognized by R in the .csv format. Therefore, dollar signs, hyphens, commas, and extra spaces are removed from dollar amounts. The fields that were fixed are now grouped together to obtain an R compatible data set.

1.2.2 Calculated Variables

All date fields are converted to days following January 1, 1900 to enable Excel compatibility in the future. After date fields have been converted to numeric, age is calculated as the difference between injury date and birth date. Similary, tenure equals the difference between injury date and date of first employment. Report lag is computed as claim filing date minus injury date. Although these calculations, being linear combinations of dates, are not required for training and testing the data, such calculations are important in the context of interpretability and predicting claim severity of new cases. Any date, being time dependent, is unique and cannot be observed in the future. Differences between dates, as discussed before, are observable in the future and have predictive value.

1.2.3 Binary Indicator Assignment

The litigation date appeared in relatively few records. Sparsity of this data field warrants the creation of a new field which considers litigation as a binary variable based on the presence or absence of litigation date. The same method used with litigation date variable was applied to attorney representation variable. Since the goal of this research project is to predict claims that would ultimately become severe, the data set was narrowed down to only those records which had a closed status and with zero reserve amount to ensure that no development was anticipated.

1.2.4 Topic Modelling

The data set has fields containing textual data which cannot be processed directly by machine learning algorithms. Such fields are modelled using Correlated Topics Model [12], a topic modelling algorithm. Each textual field is assigned a topic which could then be processed in machine learning.

1.3 Severity Classification

Typical loss runs of self insured entities consist of several data fields, including incurred loss, paid loss, and reserves. A dichotomous outcome variable indicating a severe or non-severe claim was desired. As recommended by workers' compensation actuaries, claims with incurred loss exceeding the 95th percentile of losses in the data set were marked as severe [11]. After this new field, named Severe, is created the incurred loss field is removed. Thus 95% of claim records were classified as non-severe and 5% of claim records were classified as severe.

1.4 Algorithms for Severity Classification

1.4.1 Support Vector Machines

Two machine learning algorithms - Support Vector Machines and Regularized Logistic Regression - were used in this research. Support vector machines have been proven for classification purposes due to their ability to map data into a higher dimensional feature space where classes become easily separable. One key advantage of SVMs is that the mapping to higher dimensional space is not explicitly known. Instead the algorithm is implemented using a "kernel trick',' whereby the inner product of mapped data is expressed as a kernel function. The learning is done in the feature space to determine the optimal hyperplane separating the classes, which will then be used for classifying new cases [1].

1.4.2 Regularized Logistic Regression

Regularized logistic regression utilizes a binomial loglikelihood using the logistic function with a combination of l_1 (LASSO), and/or l_2 (Ridge) penalties. When only the LASSO is utilized, the algorithm tends to retain only one among the coefficients of a group of correlated predictors [17]. This technique enables the fast detection of group of most essential predictors. On the other hand, a combination of LASSO and Ridge may be necessary for better predictive accuracy in a binary classification problem, as was the case in this research.

1.5 Cost Containment Strategies

Once a new case has been labelled as severe, cost control measures can be implemented effectively. Early intervention, claim management, and return to work programs could reduce WC costs significantly. Case management, patient management, prescription drug screening, and medical bill review ensure injured workers receive timely, medi-

Table 1: Confusion Matrix Variables

	Predicted N	Predicted Y
Actual N	True Negatives (TN)	False Positives (FP)
Actual Y	False Negatives (FN)	True Positives (TP)

cally necessary, and cost effective care. Healthcare providers who are experienced in treating WC cases, and who are aware of employee job responsibilites and alternative/light duty arrangements available can help injured workers return to work during the recovery period. Machine learning algorithms can predict severity outcomes and exemplify key correlations that exist between the predictors and outcome to help drive administrative and engineering controls.

1.6 Efficiency Metrics

1.6.1 Confusion Matrix-based Metrics

To measure the efficiency of an algorithm 5 metrics, namely: Sensitivity, Specificity, Positive Predictive Value, Negative Predictive Value, and Accuracy as defined by Kabacoff were considered [4]; see Table 1 for a description of the confusion matrix variables.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+ FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN + FN}}$$

$$\text{Positive Predictive Value (PPV)} = \frac{\text{TP}}{\text{FP + TP}}$$

$$\text{Negative Predictive Value (NPV)} = \frac{\text{TN}}{\text{FN + TN}}$$

$$\text{Accuracy} = \frac{\text{TN + TP}}{\text{TN+FP+FN+TP}}$$

The above metrics vary with algorithm chosen and parameters selected. An algorithm or parameter selection cannot be done based on these 5 metrics alone. A more appropriate metric based on profitability is discussed in the next sub-section.

1.6.2 Profitability-Based Efficiency Metric

When analyzing the efficiency of a machine learning algorithm, at first glance, one feels the need to increase sensitivity to be able to predict many of the severe claims correctly. Both Support Vector Machines, and Regularized Logistic Regression allow parameter variation in order to change sensitivity. If parameters are changed to increase sensitivity, it is possible that more of the not severe claims will be classified as severe, thereby resulting in a decrease in specificity.

The Tail Value at Risk of a random variable at the 95th percentile, denoted by $TVaR_{.95}$, is defined as the expected value of all claims valued at or above the 95th percentile [19]. In preparation of calculating the new metric, estimates of mean claim cost of severe claims, $\widehat{TVaR_{.95}}$, mean claim cost, $\overline{X}$, and the mean cost of non-severe claims, $\widehat{NS}$ are calculated from the data. The cost of early intervention was estimated as 10% of $\overline{X}$, following the recommendation of experts familiar with loss run data and third party administration [11]. The benefit of early intervention was estimated as a 50% reduction in ultimate claim cost [21]; let S denote the savings from early intervention effort. Then S is given by,

$$S = (.5) \times (TP \times \widehat{TVaR_{.95}} + FP \times \widehat{NS})$$

TP and FP claims were worked on resulting in 50% savings in the severe and non-severe cost category, respectively. Let C denote the costs associated with early intervention. Then C is given by,

$$C = (TP + FP - FN) \times (.1) \times \overline{X}$$

TP and FP claim handling costs incurred are partially offset by the cost reduction from not handling FN misclassified claims. Let MS denote the savings not realized

due to misclassification of severe claims. Then MS is given by,

$$MS = FN \times (.5) \times \widehat{TVaR}_{.95}$$

Since FN claims are truly severe claims, MS serves as a penalizing term for misclassification. The profitability index Π of an algorithm parameter set $\mathbf{p}$ may be defined as,

$$\Pi(\mathbf{p}) := S - C - MS$$

An optimal algorithm parameter set is successful at classifying more of the severe claims correctly, while ensuring that relatively few non-severe claims are misclassified. The metric Π allows a better comparison between parameter sets having similar values of metrics defined in 1.6.1.

CHAPTER 2

TOPIC MODELLING

Workers' compensation loss run data typically includes textual variables such as incident description or injury description. Verbal descriptions have high cardinality due to each record's textual entry being unique. Using such a field directly results in an overfitted model which lacks predictive value. To overcome this obstacle a generative topic model is used to assign a topic to replace each textual data element. This model assumes that a document is generated based on latent underlying topics, each of which has its own term distribution. In this research the Correlated Topic Model (CTM) was utilized which allows correlation between possible topics for a document.

2.1 Model Specification

Consider a document, $w = (w_1, \cdots, w_N)$, containing N words, in a corpus D. The words are drawn from a vocabulary containing V words. The number of topics K is specified beforehand. A random variable follows the categorical distribution if it has a multinomial distribution with n=1 (one trial). CTM assumes the following 3-step generative process for words in a document [12].

- For each topic z_k determine its categorical word distribution parameter β_{z_k},

$$\beta_{z_k} \sim Dirichlet(\delta)$$

  $\delta \in \mathbb{R}^{+V}, \beta_{z_k} \in \mathbb{R}^V, \beta := (\beta_{z_k}) \in \mathbb{R}^{V \times K}$

- The topic proportions θ of each document is determined from η,

$$\eta \sim \mathcal{N}(\mu, \Sigma)$$

  $\theta \in \mathbb{R}^K, \eta \in \mathbb{R}^{K-1}, \Sigma \in \mathbb{R}^{(K-1)\times(K-1)}, \tilde{\eta}^T := (\eta^T, 0), \theta_k = \frac{\exp(\tilde{\eta_k})}{\sum_{j=1}^{K} \exp(\tilde{\eta_j})}$

- For each word w_i in the document

- Select a topic z_i, $z_i \sim Categorical(\theta)$
- Select a word w_i, $w_i \sim Categorical(\beta_{z_i})$

2.2 Estimating Model Parameters μ, Σ, and β

The sum of log-likelihoods of all documents is minimized by varying parameters μ, Σ, and β. Log-likelihood of one document $w \in D$ is as follows.

$$l(\mu, \Sigma, \beta) = log(p(w|\mu, \Sigma, \beta)) = log \int \{\sum_{z} [\prod_{i=1}^{N} p(w_i|z_i, \beta) p(z_i|\theta)]\} p(\theta|\mu, \Sigma) d\theta$$

A Variational Expectation Maximization (VEM) algorithm [14] as outlined in the following steps is used to estimate model parameters instead of the usual Expectation Maximization (EM) algorithm due to the intractable nature of integrals involved.

- Posterior distribution $p(\eta, z|w, \mu, \Sigma, \beta)$ is replaced by a variational distribution $q(\eta, z|\lambda, \nu^2, \phi)$
- Variational Parameters based on Kullback-Leibler (KL) divergence is determined:

$$(\lambda^*, \nu^*, \phi^*) = \arg \min_{(\lambda,\nu,\phi)} D_{KL}(q(\eta, z|\lambda, \nu^2, \phi) \| p(\eta, z|w, \mu, \Sigma, \beta))$$

$$q(\eta, z|\lambda, \nu^2, \phi) = \prod_{k=1}^{K-1} q_1(\eta_k|\lambda_k, {\nu_k}^2) \prod_{i=1}^{N} q_2(z_i|\phi_i)$$

where $q_1()$ is a univariate Gaussian distribution with mean λ_k and variance ${\nu_k}^2$, and $q_2()$ denotes a categorical distribution with parameters ϕ_i.

$$log(p(w|\mu, \Sigma, \beta)) = L(\lambda, \nu, \phi) + D_{KL}(q(\eta, z|\lambda, \nu^2, \phi) \| p(\eta, z|w, \mu, \Sigma, \beta))$$

$$L(\lambda, \nu, \phi) := E_q[log(p(\eta, z, w|\mu, \Sigma, \beta))] - E_q[log(q(\eta, z))]$$

Maximizing L with respect to variational parameters is equivalent to minimizing KL divergence between functions p and q.

- E-Step: For each document find optimal value of λ, ν, and ϕ.
- M-Step: Maximize resulting lower bound of log-likelihood with respect to η, Σ, and β.

2.3 Document Term Matrix

The R package "topicmodels" accepts a document term matrix - matrix having rows and columns representing records and words - for fitting a topic model. In order to create a document term matrix from a text field, the R package "tm" is used. The vocabulary is usually not known beforehand, but is extracted from pertinent data field. All words in the field are initially considered. Several filters such as stemming, punctuation removal, case lowering, minimum word length, etc. are applied to the initial set of words. Furthermore, a term-frequency-Inverse document frequency measure helps reduce the vocabulary to only those words occurring more frequently and those occuring in not many documents based on acceptable threshold. The term-frequency-Inverse document frequency of a term, t, in a corpus, D, could be defined as:

$$tfidf(t) := f_t * log \frac{|D|}{|d \in D : t \in D|}$$

Whereas, first multiplicand (number of times term t occurs in the corpus) gives more weight to terms having high frequency in the corpus, the second term gives more weight to terms that are not found in many documents. While selecting the threshold for eliminating terms with low $tfidf$ values, care was taken to have this less than or equal to the median [12] of values among the terms, but not as large to prevent the algorithm from assigning a single topic to each document.

2.4 Application of CTM to Incident Descriptions

After a document term matrix with a vocabulary of suitable size is created, it is fitted to the CTM model using the R-package "topicmodels" by calling the "CTM" function and providing necessary parameters. The number of topics was selected to be 30 as seen in scholarly literature [12]. A cross validation method may be utilized to calculate the optimal topic number for a given data set. To investigate the effectiveness of the algorithm, the "terms" function is invoked to view the five most frequent terms under each topic. In order to obtain reasonable groupings of terms under topics one may need to change the number of topics. As the output of this function is a column of topic numbers, the word "topic" is appended to each topic number to ensure proper type recognition. Moreover, the incident description field is replaced by the new topic field. Other textual description fields are identified and replaced with corresponding topic fields.

CHAPTER 3

SUPPORT VECTOR MACHINES

The R-package **kernlab** contains the *ksvm* function, an implementation of support vector classification [2]. Support vector machines(SVM) have gained popularity due to their efficiency and simplicity in dealing with classification problems. SVMs attempt to separate data points using an optimal hyperplane that maximizes the separation between support vector margins of either class [3]. In cases where data points are not linearly separable, they are mapped to a higher dimensional space to enable linear separation. When data classes overlap across the decision surface a cost parameter is included in the method to penalize misclassification. The problem is solved as a constrained quadratic optimization problem.

3.1 SVM Theory

Consider the hyperplane (decision function),

$$f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b = 0$$

where Φ is a mapping from the data dimension to a higher dimension. The primal optimization problem involving a soft margin (one that allows overlap across decision surface) for a training set with m points has the form [1]:

$$\text{Minimize} \quad t(\mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m} \xi_i$$

$$\text{Subject to} \quad y_i(\langle \Phi(x_i), \mathbf{w} \rangle + b) \geqslant 1 - \xi_i \ (i = 1, ..., m)$$

$$\text{and} \quad \xi_i \geqslant 0 \quad (i = 1, ..., m)$$

where C is the cost parameter, $y_i \in \{+1, -1\}$. By the method of Lagrange multipliers we get

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \Phi(x_i)$$

Restating as a quadratic optimization problem [1],

$$\text{Maximize} \quad W(\alpha) = \alpha^T \mathbf{1} - \frac{1}{2}\alpha^T H \alpha$$

$$\text{Subject to} \quad 0 \leqslant \alpha_i \leqslant \frac{C}{m}, \quad i \in \{1, ..., m\}$$

$$\text{and} \quad \alpha^T \mathbf{y} = 0$$

where $\alpha \in \mathbb{R}^m$ is a vector of Lagrange multipliers, $H_{i,j}$=$y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{y}$ is the vector of class labels, and k is the Gaussian radial basis function kernel given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

As seen in the kernel formula, instead of calculating $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$ separately and then obtaining the inner product, kernel function ("kernel trick") allows direct substitution of $\mathbf{x}_i$ and $\mathbf{x}_j$ to calculate entries in the H matrix. Solving the optimization problem yields the vector α of lagrange multipliers. A non-zero value for α_i indicates that the corresponding data point is a support vector.

3.2 SVM Parameter Selection

Parameters C and γ are specified by the user to change the complexity of the model. Generally, a higher cost parameter results in the algorithm creating a more complex model due to the higher cost of a misclassification error. However, a higher cost model, due to overfitting, may have poor generalization ability resulting in poor prediction when used with test data.

The package **e1071** has built-in parameter selection capability. In this research project this function could not be used due to the high volume of data. An alternate approach was to code in better error handling methods while using modified 10-fold cross validation to compute the five confusion matrix based metrics discussed in Section 1.6.1. Furthermore, the profitability based metric discussed in Section 1.6.2 was calculated to provide another layer of selectivity. By considering all six metrics one can find the best parameter set for the SVM algorithm.

3.3 Addressing Assymetric Class Distribution in Data

In section 1.3, severe claims were defined as those at the 95^{th} percentile or above. Due to this definition, the data set has a disproportionate number of severe and non-severe cases. Package **e1071** has provision for adding class weights in the *svm* function call [2]. For the data set being studied, class weighting resulted in a sensitivity of 43%.

3.3.1 Symmetry Factor & Proportion of Predictions

As advised by Qiang Wu [16], an analysis was conducted to measure the effect of class symmetry on efficiency metrics. The symmetry factor, SF of a data set may be defined as,

$$SF := \frac{\text{Number of Non Severe Cases}}{\text{Number of Severe Cases}}$$

The results of the analysis are given in Table 2. In Table 2, the leftmost column indicates how many of the five predictive models derived from balanced data, used for this study gave a severe outcome; the top row has different values for SF. The table gives Π values associated with each combination in millions of dollars. The metric, Π was optimal when $SF \approx 1$, i.e. when the training data contained an equal number of severe and non-severe cases. Moreover, when several balanced groupings were trained and each of those models were used to predict outcomes for test data, as long as at least one model classified as severe, the final outcomes had maximum chance of being severe. Requiring more models to indicate a severe classification resulted in decrease in sensitivity and Π. The above results warranted a deviation from the conventional approach to running SVM algorithm on data. A modified version of 10-fold cross validation is discussed in section 3.3.3.

3.3.2 Data Preparation For Cross Validation

It was suspected that there might be clusters of data from either class throughout the data set. A random sampling method which shuffles the data records was employed

Table 2: Severe Classification vs. Class Balance

# of Ys	1/3	1/2	1	2	3
> 0	1.92	1.97	1.84	1.53	1.37
> 1	1.86	1.87	1.70	1.40	1.29
> 2	1.84	1.83	1.54	1.39	1.27
> 3	1.75	1.65	1.42	1.10	1.04
> 4	1.56	1.40	1.12	.72	.58

to ensure a more uniform distribution of the classes. In preparation for 10-fold cross validation [8], the data was partitioned into 10 equal non-overlapping parts. Nine parts would serve as training data while one part would be the test data. The final calculated metric values are the average of the 10 algorithm runs.

3.3.3 Modified 10-Fold Cross Validation

As indicated in the previous paragraph, *svm* algorithm is not directly run on the nine parts of training data. Instead, a stratified cross validation methodology is utilized. All severe classified records are extracted and paired with each of the 19 (on average assuming uniform distribution of classes) possible non-severe classified record sets. The algorithm is trained using each of these 19 symmetric data sets (each having an equal number of severe and non-severe classified records), and a prediction is performed on the held out training data. If at least one of 19 predictions indicated a severe classification, the final classification assignment was severe. Thus this modified 10-fold cross-validation requires 190 train-predict cycles to compute the final average efficiency metric values.

3.3.4 Multiple Modified 10-Fold Cross Validation

The procedure outlined in section 3.3.3 gives a single set of efficiency metric values. In order to gauge the reliability of the algorithm, more sets of efficiency metrics

are desired. When comparing two different algorithms, the data set is shuffled after seeding the random number generator with the same value to aid proper comparison. In this case where we need multiple sets of efficiency metrics, the random number generator is seeded with different, but known values, and the process in section 3.3.3 is repeated to get more sets of efficiency metrics.

3.4 Prediction of Outcomes from Client's Perspective

The methods discussed thus far involved training and prediction from an R programmer's perspective. Once a model is trained it needs to be made available for prediction to individuals who may not have a working knowledge of R programming. Also, in some cases businesses are not willing to disclose proprietary information to vendors. In response to a question posed by Tim Coomer & Lu Xiong [20], an investigation was conducted to determine whether a set of coefficients or a coefficient matrix could be extracted to facilitate prediction in an online platform that does not support R-packages. The function *svm* in package **e1071** does not allow extraction of certain attributes required in this scenario. The function *ksvm* in package **kernlab** has provision for extracting the necessary attributes from the *model* object [2]. Consider the decision function $f(\mathbf{x})$ given by,

$$f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b = \langle \sum_{i=1}^{m} \alpha_i y_i \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle + b = \sum_{i=1}^{m} \alpha_i y_i \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle + b$$

$$= \sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b = \sum_{i=1}^{m} \alpha_i y_i exp(-\gamma \|\mathbf{x}_i - \mathbf{x}\|^2) + b$$

The following steps summarize the evaluation of the decision function f for a new case $\mathbf{x}$ along with the necessary R code:

- Evaluate entries in the m×p matrix $[\mathbf{x}_i - \mathbf{x}]$ where $\mathbf{x}_i$ corresponds to each vector in the model matrix of support vectors, where m and p represents the number of support vectors and the number of fields for each support vector, respectively.

$$[\mathbf{x}_i - \mathbf{x}] \equiv X <- sweep(model@xmatrix[[1]], 2, \mathbf{x}, "-")$$

Matrix of support vectors is represented by $model@xmatrix[[1]]$. The *sweep* function subtracts $\mathbf{x}$ from each support vector, $\mathbf{x}_i$.

- Calculate entries in the vector, $[\|\mathbf{x}_i - \mathbf{x}\|^2]$ of size m,

$$[\|\mathbf{x}_i - \mathbf{x}\|^2] \equiv \mathbf{Y} <- rowSums(X * X)$$

This steps performs norm squared as an inner product. The * operation for matrices, which performs product of elements having equal indices, followed by $rowSums$ function achieves the goal.

- Determine entries in the vector, $[exp(-\gamma * \|\mathbf{x}_i - \mathbf{x}\|^2)]$ of size m of evaluated kernel functions, . In the $ksvm$ function γ is called σ.

$$\gamma <- model@kernelf@kpar\$sigma$$

$$[exp(-\gamma * \|\mathbf{x}_i - \mathbf{x}\|^2)] \equiv \mathbf{K} <- lapply(\mathbf{Y}, function(x)\ exp(-\gamma * x))$$

The *lapply* function replaces each element, x, in the argument object $\mathbf{Y}$ and replaces it with $exp(-\gamma * x))$ and assigns the results to an object $\mathbf{K}$

- Evaluate the decision function $f(\mathbf{x})$. In the function $ksvm$, b represents the negative intercept.

$$b <- (-model@b)$$

$$f(\mathbf{x}) \equiv Decision <- model@coef[[1]]\% * \%\mathbf{K} + b$$

The vector $[\alpha_i y_i]$ of size m is stored in $model@coef[[1]]$. The operator $\% * \%$ performs the inner product of the two vectors.

The severity classification of a given new case is determined as follows.

$$f(\mathbf{x}) \geqslant 1 \Rightarrow \text{ Severe}$$

$$f(\mathbf{x}) \leqslant -1 \Rightarrow \text{ Not Severe}$$

CHAPTER 4

ALGORITHM & VARIABLE SELECTION

This chapter discusses the relative benefits of five different algorithms that are widely used for classification problems as well as the method of regularized logistic regression with lasso penalty in order to identify the best predictors in the data.

4.1 Algorithm Comparison

A discussion of SVM algorithm for severity classification was provided in chapter 3. Five different types of algorithms - Support Vector Machines (SVM), Naive Bayes (NB), CART, Logistic Regression (LR), and Regularized Logistic Regression (RLR) were investigated for their suitability in severity classification. Table 3 summarizes the performance metrics associated with each of these algorithms. SVM was selected for this research due to its provision for varying cost and gamma parameters, and also due its high overall accuracy. This parameter variability is more important when the pi-metric is considered. Changes in savings from early intervention, and early intervention costs, may necessitate change in algorithm parameters to ensure maximum profit.

While studying logistic regression algorithm it was observed that prediction was not always possible due to the high cardinality of certain variables. When data was split into test and training data, several variables had records with categories in the test set but not in the training set and vice versa. The random forest algorithm was unusable due to its incapability to address high cardinality.

4.2 Variable Selection

To enable efficient and feasible implementation of predictive algorithms one needs to identify predictors in the data-set which have the best predictive ability. As discussed

in the previous section, the logistic regression algorithm encountered problems, and thus could not be used for step-wise dimensionality reduction.

4.2.1 RLR for Dimension Reduction

Regularized logistic regression with the lasso, or l_1 penalty using a cyclical coordinate descent optimization method is known for fast executability [17]. The lasso penalty shrinks coefficients of correlated variables so that only one of them will finally have a non-zero coefficient.

RLR with the Lasso (l_1) penalty minimizes the following function over $(\beta_0, \beta) \in \mathbb{R}^{p+1}$, where p is the dimension of the data [17].

$$l(\beta_0, \beta) = \frac{1}{N} \sum_{i=1}^{N} y_i.(\beta_0 + x_i^T \beta) - \log(1 + \exp(\beta_0 + x_i^T \beta)) - \lambda \|\beta\|_{l_1}$$

The value of λ that minimizes cross validated error for the model is selected using "cv.glmnet" function [18]. The "predict" function uses the model coefficients associated with this λ to perform prediction on test data.

Although the model matrix initially consisted of over 3065 columns, after regularized logistic regression only 158 columns (predictors with non-zero coefficients) remained. This corresponds to 28 predictors in the initial data set, and only 16 predictors in the dimension reduced data set. Among this new set of predictors were lag time of reporting, injury causing object, litigation status, State of jurisdiction, cause of injury, loss description topic, and target body part.

In addition to helping find the most effective variables, this algorithm yields the subset of categories of a variable with the strongest influence on severity outcomes. For implementing a predictive analytics solution online, a smaller set of variables and a smaller subset of categories in those variables reduces the time a client has to spend entering data or selecting categories for a new case for which a severity outcome is desired.

Table 3: Comparison of Algorithms

Algorithm	Sensitivity	Specificity	PPV	NPV	Accuracy	Π
SVM	.98	.67	.14	1.0	.69	$585,111
NB	.97	.63	.12	1.0	.65	$568,466
CART	1.0	.48	.09	1.0	.51	$573,231
RLR	.93	.79	.19	1.0	.80	$539,157

CHAPTER 5

RESULTS, CONCLUSION & RECOMMENDATION

5.1 Discussion of Results

The main purpose of this research was to predict severe outcomes. Existing methods of prediction had predictive accuracy of 20% or less. Also some variables were not effectively used as predictors. The topic modelling algorithm enabled classification of textual data into categories. Many of these categories had significant predictive content. A method of class balancing was utilized which improved predictive accuracy to more than 80%. Class balancing resulted in several trained models and their corresponding predictions on test data. As long as one model predicted an outcome as severe, the test data had the most chance of being severe. To enable implementation of SVM in an online platform that does not support R programs, model coefficients and support vectors were extracted and manual calculations were performed to predict outcomes. These outcomes were all cross checked with algorithm prediction on the test set. In order to identify variables which had the strongest influence on severity outcomes, the initial variable set was reduced to a smaller set using RLR with the lasso penalty.

5.2 Conclusion

Loss run data from self-insured entities have significant informative content which when properly extracted using machine learning or data mining algorithms could be used to build predictive analytic solutions for those clients. Textual data in descriptive fields can be effectively utilized to empower predictive models. Data in loss runs, by nature are assymetric between severe and non-severe claims, resulting in poor predictive results. Balancing techniques, when used properly, support the extraction

of predictive knowledge to increase accuracy. Data dimension reduction can be accomplished by fast executable coefficient shrinkage methods such as RLR with lasso penalty.

5.3 Recommendations

This research studied data specific to a single client, to enable prediction of future outcomes of that client. A more universal model based on variables which are common across industries and clients will enable higher versatility of the predictive model. Also SVM radial basis kernel was used as an investigative tool. More sophisticated kernels could improve performance. A better system of quantifying the reliability of a model is desired. An improved metric derived from a confusion matrix that allows cross-client algorithm comparison will aid in the development of a model that is generalizable across multiple client data.

BIBLIOGRAPHY

[1] A. Karatzoglou, D. Meyer, K Hornik, *Support Vector Machines in R*, Journal of Statistical Software, **15**(9),(2006)

[2] A. Karatzoglou, A. Smola, K. Hornik,*Support Vector Machines*, The Comprehensive R Archive Network, https://cran.r-project.org/web/packages/kernlab/kernlab.pdf, (2016)

[3] C. Cortes, V. Vapnik, *Support Vector Networks*, Machine Learning, Kluwer Academic Publishers, 1995, pp. 273-297

[4] R. Kabacoff, *R in Action*, Manning Publications Co., 2015, pp. 569-575.

[5] I. Sengupta, M. Baldwin, *Workers' Compensation: Benefits, Coverage, and Costs, 2013*, [https://www.nasi.org/research/2015/report-workers-compensation-benefits-coverage-costs-2013]

[6] K. Antonello, *State of the Line*, NCCI's 2015 Annual Issues Symposium, https://www.ncci.com/Articles/Documents, (2015)

[7] A. Lipold, *Workers' Comp Predictive Modeling Comes of Age*, American Academy of Actuaries: Contingencies, 2012, pp. 34-38.

[8] P. Refaeilzadeh, L. Tang, H. Liu, *Cross-Validation*, Encyclopedia of Database Systems, 2009, pp. 532-538.

[9] K. Finn, *Predictive Modeling Improves Claim Outcomes While Lowering Costs*, Perspectives, The Hartford, 2013 p 23.

[10] E. Boone, *Changing The Game in Workers Compensation*, RN Magazine, The Rough Notes Company, Inc., 2011, p.78

[11] A. Rhodes, *Sigma Archives*, SIGMA Actuarial Consulting Group, Inc, 2016

[12] B. Grun, K Hornik, *topicmodels: An R Package for Fitting Topic Models*, Journal of Statistical Software, 2011

[13] D. Blei, J. Lafferty, *A Correlated Topic Model of Science*, The Annals of Applied Statistics, 2007, pp. 17-35

[14] M. Wainwright, M. Jordan, *Graphical Models, Exponential Families, and Variational Inference*, Foundations and Trends in Machine Learning, 2008, pp 1-305

[15] D. Blei, J. Lafferty, *Topic Models*, Text Mining: Classification, Clustering, and Applications, Chapman and Hall CRC Press, 2009, p11

[16] Q. Wu, *Personal Communication*, Middle Tennessee State University, 2016

[17] J. Friedman, T. Hastie, R. Tibshirani, *Regularization Paths for Generalized Linear Models via Coordinate Descent*, Journal of Statistical Software **33**(1), (2010)

[18] J. Friedman, T. Hastie, R. Tibshirani, *Package 'glmnet'*, The Comprehensive R Archive Network, https://cran.r-project.org/web/packages/glmnet/glmnet.pdf, 2009

[19] S. Klugman, H. Panjer, G. Willmot, *Loss Models: From Data to Decisions*, Wiley Publishing Co., (2012)

[20] T. Coomer, L. Xiong, *Sigma Archives*, SIGMA Actuarial Consulting Group, Inc, 2016

[21] Corvel, *A Patient Centered Approach: The Value of Early Intervention*, Focus on Claims Management, www.riskandinsurance.com/wp-content/uploads/2015/09, CorVel Corporation, (2013)