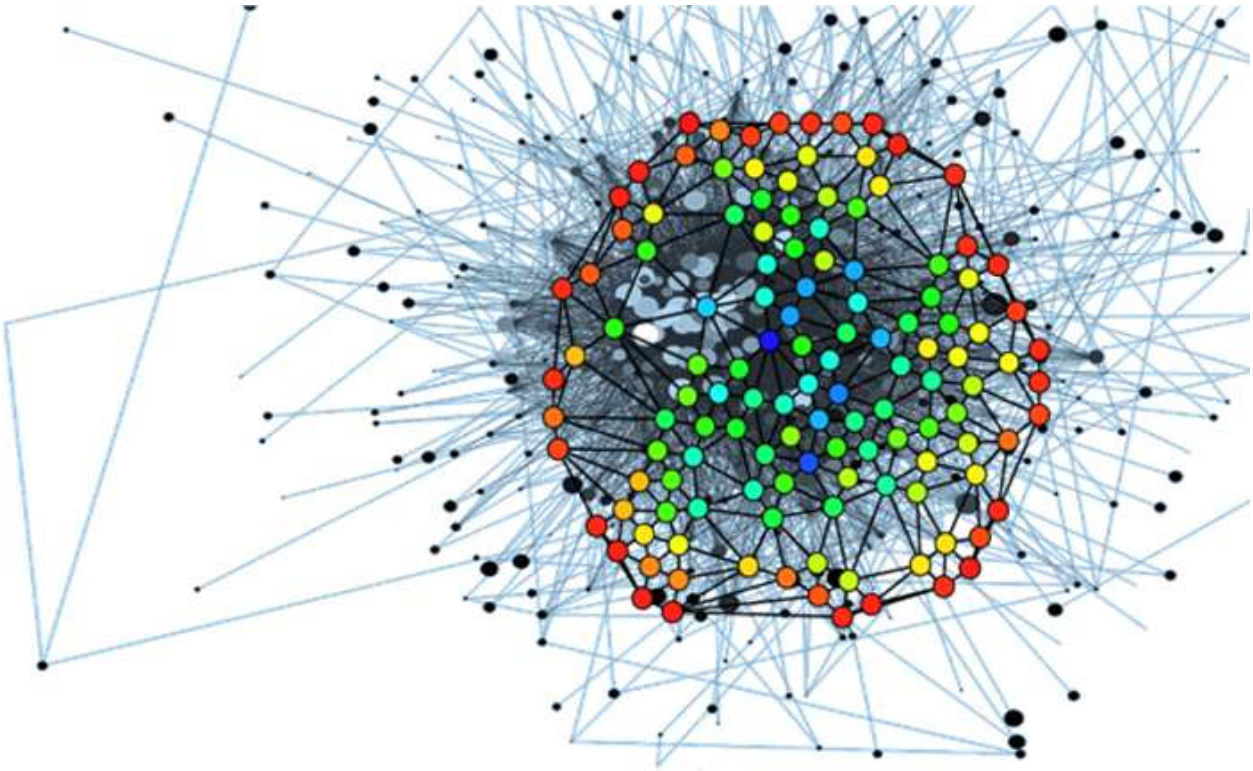
A MACHINE LEARNING APPROACH TO PREDICTING EMPLOYMENT PRACTICE LIABILITY CLAIMS

Identifying Companies Most Likely to Get Sued



Author: Christopher Allen Cirelli
Date: 11.28.2017

CONTENTS

I. INTRODUCTION

The objective of this study is to test to what extent Machine Learning algorithms can identify companies most likely to incur Employment Practice Liability (EPL) lawsuit using a simple feature and claims dataset. Employment Practice Liability arises from US Labor Law, which regulates how employers interact with their employees. Employers may be held liable for certain wrongful acts, including, but not limited to, wrongful termination, sexual harassment, discrimination, invasion of privacy, false imprisonment, breach of contract, emotional distress, and wage and hour law violations. These violations may carry monetary consequences. For instance, the top 10 Employment Practice settlements in 2015 totaled $296m in damages. In 2016, a single employer could expect to incur $125k in costs for a single EPL lawsuit and in the same year, there was an 11% chance that a US private company would incur a lawsuit. Companies may purchase insurance to transfer certain of these financial consequences to an insurance company. This implies that the better both corporations and insurance companies can understand the risk factor that give rise to EPL violations, and by extension those companies more likely to incur a lawsuit, the more efficiently both parties may allocate capital to this issue. Therefore, the objective of this study will be to train and test various Machine Learning algorithms in hopes that the findings will prove useful for both providers and users of insurance alike.

The approach to our study includes three phases. Phase I will consist of identifying and preparing our dataset. The objective of this phase is to identify the best source of data for our classification models, the optimal features to include, and the presentation of the target feature. Phase I will conclude with a look at our Analytics Based Table (ABT), which will be the basis for the data exploration and machine learning phases of our study.

Phase II will consist of the exploration and analysis of our dataset. We will utilize statistical tools to gain a better understanding for central tendencies and distribution of our data, to identify correlations between features and between features and our target variable. Phase II will conclude with a discussion of our key observations and how they may be used for the testing and training phase of our study.

Phase III will consist of the implementation and testing of the machine learning algorithms. Three such algorithms were selected for the study. These include the Decision Tree, Nearest Neighbor Classifier and Naïve Bayes classifier. Prior to implementation and training, a description of each model, its theoretical approach to learning and strengths and weaknesses will be discuss. Thereafter, we will train and test our models and analyze the results.

Our study will conclude with a discussion of key observations derived from the machine learning models, what we did and did not achieve, which models performed the best, and why. In addition, the author will propose some suggestions for refining the model through building a more robust dataset.

II. DATA SOURCES & PREPARATION

As stated in the preceding section, the purpose of this study is to utilize machine learning algorithms to identify (or distinguish between) companies that are most likely to incur an EPL lawsuit. This requires that we obtain information on companies that have been sued in the past. The purpose of this section is dedicated to discussing the sources of data that were obtained for this study. We will conclude this section with a discussion of the creation of the Analytics Base Table (ABT), which is a model for structuring data for machine learning applications.

Sources of Data

Despite both the incidence of EPL lawsuits and the average cost to defend them few if any public resources exist that provide categorical information on both companies sued and not for a given time period. For instance, the US Equal Employment Opportunity Commission aggregates statistics on cases it brings against companies[1]. While the database is robust in terms of the types of violations that are brought, it lacks essential data on the characteristics of the companies that were sued, including their size, number of employees and industry. While the possibility exists to create a dataset from filed lawsuits and merging this data with datasets from paid providers, it was ultimately determined that the time and monetary costs of this approach exceeded the scope of this project[2].

In light of the lack of publically available information that contained the type of aforementioned categorical information, the data for this study was sourced from an insurance company's claims and policy database. The claims information was housed in an Excel Spreadsheet and included, amongst other things, information on EPL policies dating back to 2010, individual claims, and a description of the allegations, claim counts and amounts incurred. The policy information was sourced from a CRM database, which provided information on the insurance contract and certain categorical features of the companies, including, but not limited to, revenues, employee count, state of domicile and industry.

Data Description

Data Source	Rows	Columns
Policy Data set	4,301	22
Claims Data set	14,652	20

To the right the reader will notice two tables. The first provides information on the shape of the two datasets. The first row refers to the policy and feature information drawn from the CRM dataset. The second the claims dataset drawn from the Excel spreadsheet. The claims dataset is much larger because each claim has its own row, leading to potentially multiple claims for any given policy. Therefore, part of this transformation process of merging our datasets was to separate out the EPL claims counts and aggregate them for each EPL policy.

[1] See: https://www.eeoc.gov/eeoc/statistics/index.cfm

[2] For this approach to work, one would have to aggregate lawsuits for all 50 states and Federal courts. Assuming that a unique ID could be found for the companies in the lawsuit dataset, this information would have to be merged with a third party database on private companies. Examples include Capital IQ or Data.com. That being said, these are paid subscriptions and donations were not solicited prior to starting this study.

The second table displays information on the types of data that are present in each dataset. The reader will note that there is a significant number of categorical and text features versus numerical. As we will discuss below, part of the work to create our ABT was to eliminate features that were lacked a predictive value for our study.

Salesforce

Data-type	Count
Categorical	10
Numerical	5
Text / ID	6
Date-Time	1
Grand Total	**22**

Loss Report

Data-type	Count
Text / ID	8
Date-Time	6
Categorical	4
Numerical	2
Grand Total	**20**

Data Preparation

The objective of the data preparation was to construct an ABT from the policy and claim datasets. First, the data type was confirmed for each feature to ensure that all data was of the same type[3]. Second, identify and deal with missing values. For the Revenue and Employee Count features, about 3% about 3% was missing. This information was replaced with the median value for the entire data set for both features. In addition, a significant number of industry names were missing from the policy dataset. Using the SIC Code feature, these accounts were cross-reference with a table of SIC Codes and the values replaced with the approach industry name. Third, the datasets needed to be limited by policy and claim for EPL policies. Fourth, the claims data needed to be grouped by policy number and the claim converted to a binary value. Fifth, the datasets were merged on the policy number.

The last step was to limit the dataset to those features that were hypothesized to have a predictive value relative to our target feature. These included text, ID's and dates that would provide little value to the study.

Data Analytics Table

ANALYTICS BASED TABLE (ABT)

Em	State	Broker	Industry	SIC Code	Revenues	Replace Rev(0) w-Median	Change Revenues	Employees	Replace E-Count(0) w-Median	Change Employees	Target (Claim 1/0)
0	OH	A	Consulting	8732	$3,290,706	$3,290,706	-$10,131,629	224	224	-	1
1	TX	B	Finance	7322	$15,097,958	$15,097,958	-$2,042	200	200	-	0
2	VA	C	Not For Profit	8641	$50,797,720	$50,797,720	$50,640,188	2,200	2,200	-	1
3	CA	D	Machinery	3541	$234,068,645	$234,068,645	$234,068,645	1,400	1,400	400	0
4	GA	E	Telecommunications	4833	$812,465,000	$812,465,000	$466,167,000	3,996	3,996	1,748	0
5	FL	F	Insurance	6411	$16,558,563	$16,558,563	$7,858,563	100	100	-	0
6	GA	G	Retail	5047	$79,811,630	$79,811,630	$50,511,630	210	210	-	1
348	PA	H	Mining	1099	$1,388,030,494	$1,388,030,494	$1,245,230,494	1,001	1,001	-	1

A short note on the features selected is in order. The state in which a company operates is important as EPL laws vary by jurisdiction. The broker is important as certain brokers specialize in more hard to place business. Industry and SIC Code are important as certain industries tend to exhibit a higher / lower incidence of EPL lawsuits. Revenues and employee count are proxies for the size of a corporation. The Change in Revenues and Change in Employee Count Features were derived from the year / year data and may be indicative of the health of a company at any given time. As a company's revenues fall, they tend to lay off employees, which in turn can lead to claims. The target feature is a binary representation of the lawsuit or claim. If a company had one or more claims, it was given a one. Claims incidence was not counted or summed for any given policy as the objective of the study is to identify the features that may contribute to claims as opposed to how often they may occur for a given company.

Lastly, it is worth noting that there are limitations to this dataset. While historical data would suggest that certain of these features may help us to better categorize companies more or less at risk of a lawsuit, simply by knowing a company's industry and revenue size tells us little about how they treat their employees. Therefore, the author acknowledges that there may be limitations to the prediction power of this dataset.

[3] This was not the case. In particular, the policy number contained integers and strings, all of which was converted to strings before being able to merge the two datasets.

III. DATA EXPLORATION & ANALYSIS

The objective of this section is to gain a better understanding of our dataset prior to embarking on training and testing the machine learning classifiers. We will look at measures of central tendency, dispersion, and correlation. These observations will be important to determine which machine learnings algorithms are appropriate for solving the business problem at hand.

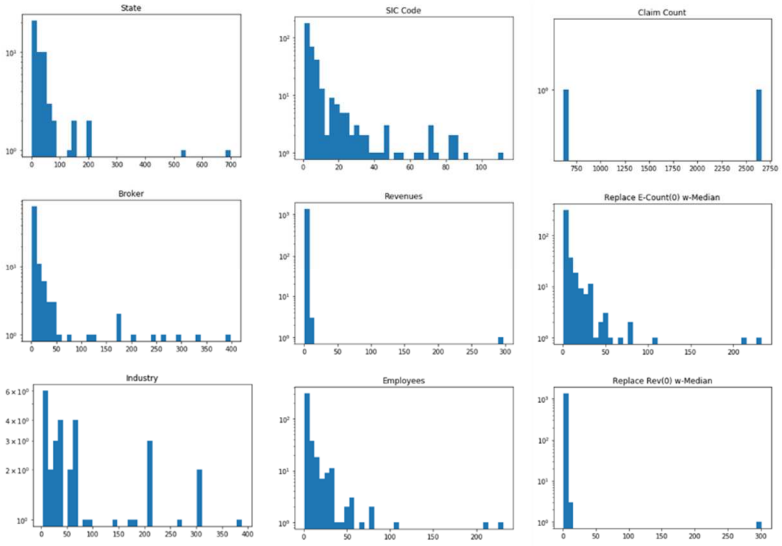
Central Tendency

To the right the reader will see a table that includes certain measures of central tendency for each numeric feature. These include mean, mode, standard deviation, and min, max and three quartile values. There are a number of key observations that were derived from this table. First, we have a small dataset. This will be important later on when determining the sizes of our train and test set sizes. Second, and simply by looking at the min and max values, we can surmise that we have large outliers for each of the numerical features in our data set. For instance, the min value for revenues is $0, with a mean of 279m and max of $56bn. We will need to consider the extent of this dispersion when constructing our machine learning models. Lastly, it is worth noting that we may have some missing data. For instance, the Change in Employees features shows zeros for the 25-75% quantiles[4].

	SIC Code	Revenues	Replace Rev(0) w-Median	Change Revenues	Employees	Replace E-Count(0) w-Median	Change Employees	Claim Count
count	3,277	3,277	3,277	3,277	3,277	3,277	3,277	3,277
mean	6841	$279,974,788	$281,074,165	136,638,270	1,204	1,209	169	19%
std	2101	$1,594,176,087	$1,593,986,745	1,194,871,078	6,304	6,303	2,128	39%
min	174	$0	$223	(4,291,621,777)	-	1	(3,000)	0
25%	5813	$746,000	$2,349,913	-	7	15	-	0
50%	7381	$12,130,156	$12,130,156	1,649,503	68	68	-	0
75%	8361	$77,517,687	$77,517,687	34,032,820	400	400	-	0
max	9999	$56,800,000,000	$56,800,000,000	56,730,100,000	127,368	127,368	50,882	1

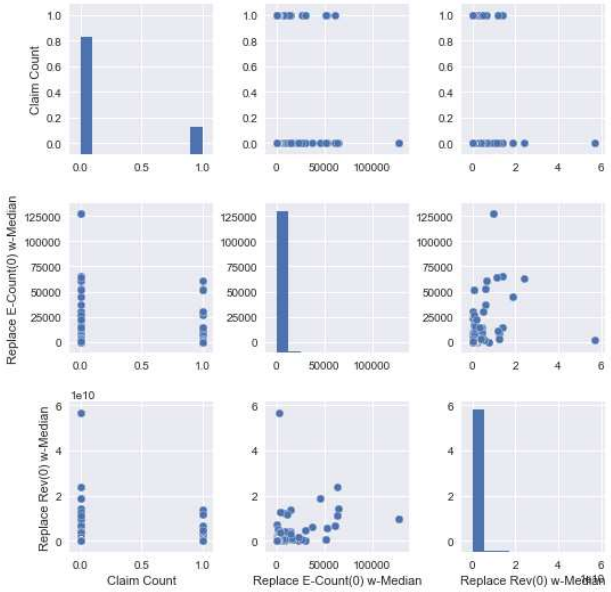
Dispersion

To the right the reader will see a series of charts represents the dispersion of each of our numerical features. These values were set to a logarithmic scale in order to better visualize their distribution. We can see from these charts that all of the features are either unimodal (tending to one extreme), or exponentially distributed. We can also see that certain features contain outliers (examples include employee count and revenues).



[4] It turns out that these values are actual zeros, i.e. the majority of companies in the study reported 0 change in employee count from one year the next and therefore, no changes were made to these values.

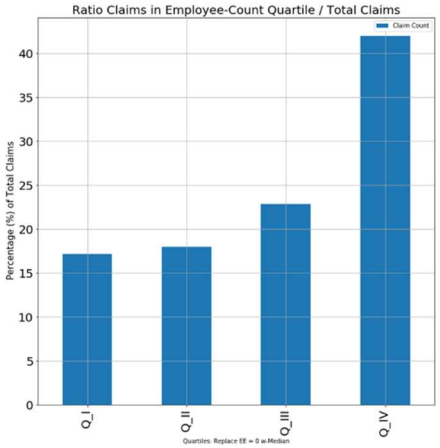
Scatter Matrix



To the right the reading will find a scatter matrix, which shows the relationship between feature values and between feature and target values. It would appear that there is a strong linear relationship between Revenues and Employee count. This would seem logical in that, but for a few industries, the more revenues a company has (i.e. its size), the more employees it is likely to have. As it relates to our target value, we have already ascertained that 18% of the companies in our dataset incurred a claim. Therefore, we should expect to see for any given feature more dots for 0 column than 1. No additional observations were made using the scatter matrix.

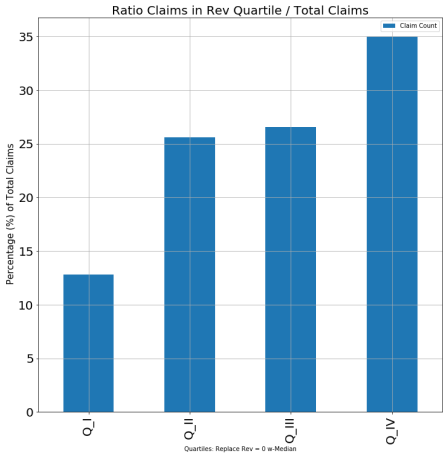
Feature Importance

In light of some of the difficulties associated with ascertaining relationships between our feature and target values using the scatter matrix, separate graphs were created for each of our features. In particular, we will look at the relationship between Employee Count, Revenues, State, Industry and ^ to Employee Count and Revenues as it relates to the prevalence of claims. Each of these graphs has a similar structure. Companies are group by a single feature, whether that be size or a sublevel within the feature, and then the percentage of companies with claims is calculated for that grouping. The y-axis represents the percentage of claims and the x-axis the feature. In turn, this should help us to better understand the relationship, or lack thereof, between our feature and target.

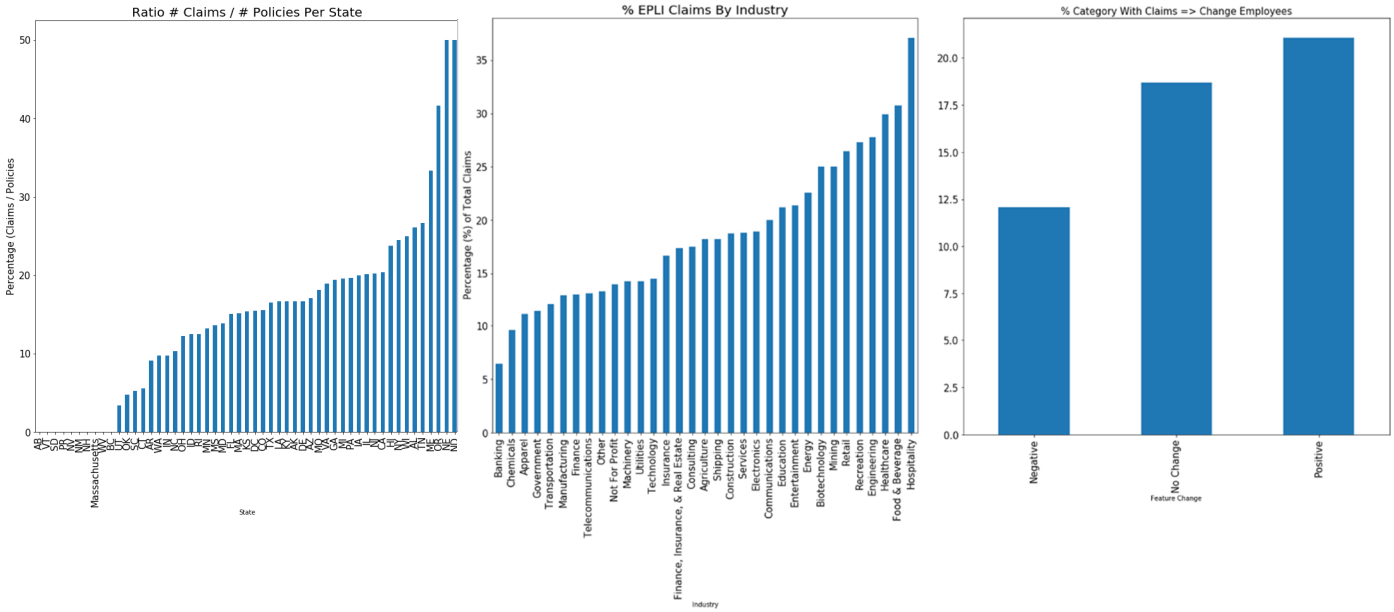


The first chart shows the relationship of claims to Employee count. Employee count has been broken up into quartiles. As we can see from the chart, there appears to be a linear relationship between employee count and the incidence of claims. This may have a simple explanation in that the more employees a company has the higher the chance it has to be sued.

The second chart shows the relationship of claims to revenues. Here we find an even strong linear relationship, in which the percentage accounts with claims increases with each successive group.



Below the reader will find three charts, each of which looks at the relationship of claims for features in our study. There are clearly certain states in which more companies have experienced claims than others. The same can be said for different industry groups. It is important to note that given the size of our data set some sub features may include only a handful of examples, which may not generalize well to large datasets.



The last relationship that we looked at was the percentage of accounts with claims vs the change in Revenues and Employee count. Here it appears that companies with increasing revenues and employee counts experienced more claims than those with no change or decreasing revenues and employee count.

Key Takeaways:

We began the investigation of our dataset with the objective of deriving information that could prove important the design, training and testing of our machine learning algorithms. We looked at measurements of central tendency, noting that certain features had large outliers. We also looked at measurements of dispersion and noted that all the features exhibited either an exponential or a unimodal distribution. We also used a scatter matrix to compare our features versus the target value. Here we noticed that a few of our features, in particular revenues and employee count, appeared to exhibit a linear relationship. Lastly, we looked at feature importance and the relationship between the subgroups of our features and the percentage of companies that had incurred claims. This purpose was to see if claims were evenly distributed amongst the subgroups. We noticed that they were not, and in the case of Revenues, Employee Count, State and Industry, the data exhibited pronounced differences between subgroups. Each of these key takeaways is important, as they will help instruct us in how to design, test and train the machine learning algorithms in the last section of our study. In the next section, we will take a brief pause in our study to discuss the basic tenants of the machine learning and the algorithms that will be used in our study.

IV. MACHINE LEARNING: 'INTRODUCTION AND APPROACH'

Thus far, in our discussion we have introduced our business problem, which is an attempt to predict companies most likely to incur an EPL lawsuit. We have discussed the importance of the source of the data for our study and its characteristics. We have also explored out data using various statistical and visual tools to gain a better understanding of the relationship between our feature and target values. Prior to moving the training and testing of our machine learning models we will discuss why machine learning is an good tool to help solve our business problem.

Why Machine Learning?

Machine learning is the process of programming computers to learn from data. There are two types of machine learning, supervised and unsupervised learning. Supervised learning requires human intervention to select and modify the algorithms. Unsupervised learning requires no human intervention and allows the algorithm to pick the features and parameters that best fit the business problem. Machine learning is particularly suited for large datasets.

Turning to insurance, insurance companies house thousands, sometimes millions of data points for current and former clients. The underwriting process largely consists of looking for similarities between the company in question and prior claims experience to form an opinion about the risk. If the client 'looks' like a lot like a companies that have been sued in the past, then either coverage is not extended or changes are made to the terms and conditions of the contract to mitigate the insurance companies exposure.

From these two very simple characterization of machine learning and insurance begins to emerge the reasoning for why machine learning is an ideal tool for insurance company underwriting problems. Insurance problems and machine learning applications share similarities in terms of the amount of data they handle. In addition, insurance underwriting and machine learning share similarities in the way that opinions or predictions are formed, i.e. by association. Therefore, it is argued that for our current business problem, which is focused on learning how to better classify companies by their propensity to incur an EPL violation, machine learning is a good tool to utilize. That said which machine learning algorithms are most appropriate for our study? Let us begin by discussing the tenants of the Decision Tree classifier.

Decision Tree

A decision tree is a flow chart like structure that includes nodes and leaves. Each node represents a test performed on a feature (dependent variable). This test consists of determining to what extend the existence or absence of this feature helps to better differentiate between those records (in our case accounts) that include and do not include the target feature (in our case, a claim). This process is known as information gain. The algorithm calculates the information gain for each feature and determines which provides for the highest gain. This feature in turn becomes what is known as the root node from which a branch is created to a second node, and the process is repeated, excluding the first feature, the algorithm attempts to find the next feature that creates the most information gain. Once fully grown, and if the classifier is successful, the structure should perfectly separate companies by a sequence of features into those that incurred and did not incur a claim.

There are both advantages and disadvantages to the Decision Tree classifier. Beginning with the latter, if grown unchecked[5], the decision tree tends to over-fit the dataset, which detracts from its ability to generalize to the test set (unknown samples). In addition, decision trees may become unstable as small variations in the dataset can create large differences in the ultimate tree structure. This could prove

[5] Hyper-parameters can be modified in order to 'prune' the tree such that it does not over-fit the test set.

challenging for a continuous model that are continuously updated with new data (think new insurance contracts as they are written and claims incurred)[6]. As for the benefits, a Decision Tree is a white-box model, which means that the construction of the tree and predictions can be visualized and verified by the user. In addition, Decision Trees can handle both numerical and categorical data, which opens up interesting possibilities for business challenges that cannot easily be boiled down to numerical representations. Lastly, Decision Trees provide for the alternation of a number of hyper-parameters, which allows the designer a wide latitude to the ultimate design of the tree.[7].

In relation to our stated business problem, the Decision Tree classifier should prove a robust model for classifying companies that are most likely to incur an EPL lawsuit. First, we are dealing with both numerical and categorical data. Secondly, our data exploration revealed relationships between our features and target groups. Third, we will be able to inspect our model to gain further insights about our data set and the relationship between our feature and target variables. In conclusion, the use of the Decision Tree classifier should provide both education and instructive in terms of solving our stated business problem.

Random Forest

The Random Tree classifier is a derivation of the Decision Tree. It uses an Ensemble Learning method to generate predictions. The process is straightforward. Multiple trees are constructed using randomly sampled training sets. A prediction is made based on the prediction from the highest number of trees. For instance, if 100 trees are grown and 51 vote for option A, then option A is the final prediction. There are multiple techniques that can be used to achieve this result. For instance, Bagging and pasting is an approach to ensemble learning in which numerous decision trees (or other classifiers) are trained on randomly selected training sets with replacement. Therefore, samples in the first tree may show up in the second to n-tree. This helps to reduce both the bias and variance of the model. Boosting is another technique to enhance the results of the Random Forest classifier. Boosting refers to an ensemble method that seeks to increase the prediction power of a given classifier by correcting areas in which it under fit the dataset. For instance, a Decision Tree can be trained to favor splitting on features that have higher weights. Given this approach, each successive iteration of the classifier should yield an increasing accuracy score. In addition to the standard Random Tree classifier, both of these methods will be test in hopes of enhancing results.

Why would Ensemble Learning in general, and the Random Forest classifier in particularly, be helpful for our study? Given a single weak predictor, for example a decision tree, and due to the law of large numbers, run numerous times on a randomized dataset, the average of the results tends to generate a more precise prediction. This is the essence of Ensemble Learning and is the foundation of both the Bagging and Random Tree algorithms. Therefore, by utilizing Ensemble Learning techniques we will attempt to increase the prediction power of our single Decision Tree experiment.

Nearest Neighbor

The Nearest Neighbor algorithm is based on the assumption that differences between feature values can be represented by a Euclidean distance. The farther two points are from one another in a Euclidean space the less they are assumed to have in common. The Nearest Neighbor algorithm makes predictions by looking at samples that were plotted nearest to our unknown sample. For instance, if the 3

[6] Random Forest is an Ensemble approach that may turn weak classifiers into strong ones using a voting technique based on the generation of multiple trees over a randomly selected dataset.

[7] Note that the discussion of the pros and cons of each model are not meant to be exhaustive. Rather, those features that are chosen are meant to highlight those that are most important to the author and to provoke though on the part of the reader.

of the 5 nearest points in a Euclidean distance to unknown sample are positive, then NN predicts that our sample must be positive. In the case of our study, if three of the five nearest neighbors to our unknown sample had claims, then the Nearest Neighbor algorithm will predict that this sample will have a claim as well.

There are a number of NN algorithms with variations to this basic approach, including, but not limited to the KD-tree, Ball-Tree, Nearest Centroid, and Nearest Shrunken centroid. Their focus is largely on limiting the computational time to generate predictions. Given the relatively small size of our dataset, these alternative approaches will not be discussed.

Assuming that companies with and without claims share similarities in terms of the features selected for our study, the Nearest Neighbor algorithm should generalize well to unknown sample sets.

Naïve Bayes

The Naïve Bayes algorithm is probabilistic classifier based on the Naïve Bayes Theorem. Conditional probabilities are generated for each feature, each level of the feature and for each value of the target variable[8]. The model uses the product of the feature and conditional probabilities to make a prediction. Based on a set of features, if the product of the conditional probabilities is higher for no-claim, then the model will predict no claim.

Unlike the Decision Tree classifier, Naïve Bayes does not include hyper-parameters. That being said, and in light of the correlation between features noted in the data exploration phase of our study, an approach of feature elimination was used to enhance the model's results. The assumption is that if features are highly correlated, they will detract from the models prediction power. Therefore, by systematically eliminating features before running the model we may arrive at a more accurate result[9].

The Naïve Bayes classifier is considered a good fit for our study as it is largely predicated on the frequency with which features appear for a given target. The higher the delineation between the features present for positive and negative targets the higher prediction power of this model.

How should the success of our model be measured?

There is a plethora of measurements used to gauge the effectiveness of Machine Learning algorithms. These include 'precision', 'accuracy', 'f1' and 'recall'. After briefly discussion, each an argument will be made for why the 'recall-score' is the most appropriate for our study.

To begin, 'precision' is a measure of how often model is correct when it predicts 'true'. Using the confusion matrix we derive the 'precision' score from the following formula True-positive / (True-positive + False Positive). Therefore, of the total predictions we made for yes, what percentage were correct. 'Accuracy' is derived from the formula (True Positive + True Negative) / N, where N equals the total number of samples predicted. Accuracy therefore is a measure of how often our model is correct. If N = 100, and TP + TN = 50, then our model accurately predicted 50% of the samples. 'Recall' is derived from the formula True Positive / (True Positive + False Negative) and is a measure of what percentage of total true values our model predicted correctly. If of 100 samples, 50 patients had the disease for which we are making predictions, and our model predicts 25 correctly, then our model has a Recall score of 50%. Lastly, f1 is the mean of our Recall and Prevision values.

[8] In the case of the EPL claims study, the conditional probability is taken for claim and no claim.

[9] In the data analysis section of our discussion, we showed that there is a high correlation between revenues and employee count, which is inconsistent with the feature independent assumption of this model.

While each of these measures are important for understanding how our model performs, only Recall gets to the center of the objective of our study, which is the accurate prediction of accounts with claims. Therefore, and while reference will be made to the other aforementioned metrics, Recall will be used as the predominant score for measuring the performance of our models.

I. DECISION TREE

The approach to Decision Tree classifier was to train the algorithm without modification, inspect the results by visualizing the tree and then graph the importance of each feature.

Decision Tree – Initial Results

The chart to the right represents the initial results of the un-modified decision tree classifier. We can see from the recall and precision scores that the classifier has likely under-fit, as opposed to over-fit, the dataset. This can be due to a myriad of issues, including the dataset itself, lack of feature selection and or tree construction.

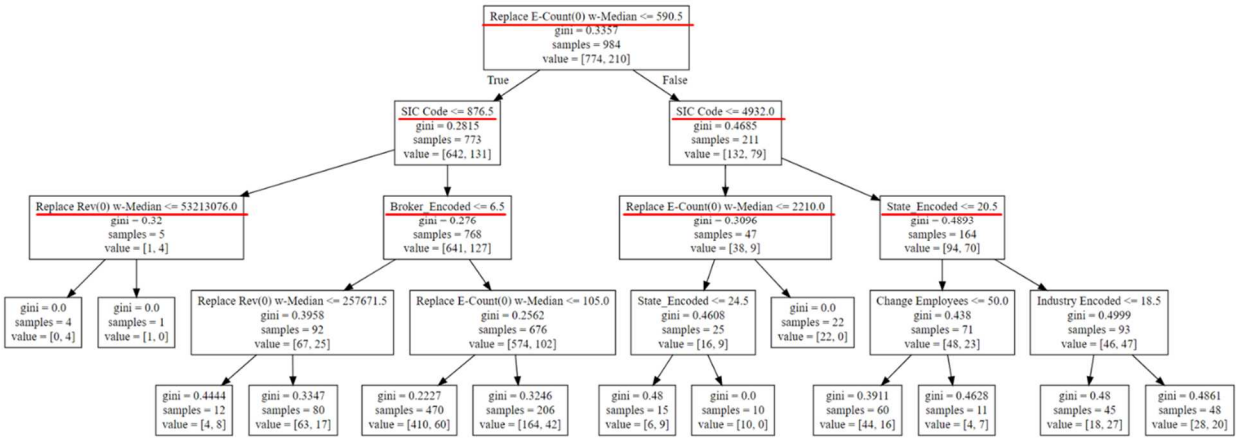
Training Results

	precision	recall	f1-score	support
0	0.93	0.98	0.95	1850
1	0.88	0.67	0.76	443
avg / total	0.92	0.92	0.92	2293

Index	Predicted NO	Predicted Yes
Actual No	1810	40
Actual Yes	144	299

Feature Analysis – Decision Tree Visualization

The below graph represents the visualization of the initial decision tree. The root node represents the employee count feature, in which the algorithm has decided to split the value by <= to 590 employees. This feature was able to differentiate between 984 features, constituting roughly 40% of our training set. This particular split generated an information gain of 0.335 or 33%. SIC Code was the feature for the second level of our decision tree. The third level of our decision tree included Revenues, Broker, Employee count and state, each generating an information gain of greater than 0.27. The construct of the below decision tree reinforces the strong relationships between these features and our target value during the data exploration phase of our study.



Feature Analysis – Feature Importance Visualization

The last step in our investigation of the basic decision tree was to calculate the percentage contribution of each feature to the 'gini' score. The chart to the right titled 'Training Set' provides a horizontal graph of these results. Here we can see in a slightly different format the important of the aforementioned features to the prediction power of our model. While most features

appear to be consistent with our original assumptions, it is odd to note that ‘Change Revenues’ is the third most import feature in terms of the model.

Test Results

The test results revealed that the classifier does a poor job of generalizing to unknown data. According to the recall score, the decision tree was able to correctly identify 34% the accounts with claims, which is insufficient for purposes of the goal of this study.

TEST-SET

	precision	recall	f1-score	support
0	0.86	0.89	0.88	810
1	0.40	0.34	0.37	174
avg / total	0.78	0.79	0.79	984

Index	Predicted NO	Predicted Yes
Actual No	720	90
Actual Yes	114	60

Key-Takeaways

It would appear that the first iteration of the decision tree does a poor job of generalizing to unknown data. Based on the graphical representation of the decision tree it appears that the model makes quick progress in differentiating between the most important features, but likely fails to find material differences between claim and non-claim companies at a more granular level. In the next section we will attempt to enhance the prediction power of our model by using Ensemble Learning and hyper-parameter selection techniques.

II. RANDOM FOREST

In this section we look at two approaches to Ensemble Learning, Bagging and Random Forest. In addition, we will use the GridSearchCV approach to arriving at an optimal hyper-parameter section for our classifier. Lastly, we will use the AdaBoostClassifier technique to attempt to enhance our recall score by focusing on the erroneous predictions of our original model.

Training Results

	precision	recall	f1-score	support
0	0.93	0.98	0.96	1861
1	0.90	0.69	0.78	432
avg / total	0.92	0.93	0.92	2293

Test Results

	precision	recall	f1-score	support
0	0.86	0.92	0.89	799
1	0.50	0.35	0.41	185
avg / total	0.79	0.81	0.80	984

Bagging:

The objective of the Bagging Classifier was to see whether we could obtain better results by creating multiple decision trees and then taking the average of the predictions. The hyper-parameters included were pruning = none, max_samples = 2,200, n_estimators = 500, Bootstrap = true.

Despite creating 500 hundred unique decision trees from a randomly selected datasets, the recall score showed little improvement, increasing by only 0.01.

Training Results

	precision	recall	f1-score	support
0	0.93	0.98	0.96	1861
1	0.88	0.70	0.78	432
avg / total	0.92	0.93	0.92	2293

Test Results

	precision	recall	f1-score	support
0	0.86	0.92	0.89	799
1	0.51	0.35	0.42	185
avg / total	0.79	0.82	0.80	984

Random Forest:

The objective of the Random Forest classifier was to obtain better results by combining the benefits of the Ensemble learning method with those of hyper-parameter selection. The first iteration of the Random tree and hyper-parameter settings of the Bagging Classifier yielded exactly the same results. This is not surprising as the techniques are the same.

GridSearchCV:

The objective of the GridSearchCV was to see if the recall score could be increased by adjusting the classifier to an optimal hyper-parameter selection. Both the results of the optimal parameters and the final recall score are displayed in the chart to the right. Unfortunately, the recall score increase by only 0.01.

Training Results
{'max_depth': 20, 'max_features': 6, 'min_samples_split': 2}

	precision	recall	f1-score	support
0	0.93	0.98	0.96	1861
1	0.89	0.70	0.78	432
avg / total	0.92	0.93	0.92	2293

Test Results
{'max_depth': 20, 'max_features': 6, 'min_samples_split': 2}

	precision	recall	f1-score	support
0	0.86	0.92	0.89	799
1	0.52	0.37	0.43	185
avg / total	0.80	0.82	0.80	984

AdaBoostClassifier:

Surprisingly, the AdaBoost Classifier applied to the Random Forest algorithm and set to the optimal hyper-parameters generated from the GridSearchCV yielded only a slight increase in the recall score. This is odd considering that the objective of this approach is to increasingly add more weight to the errors in the

underlying classifier to generate a more accurate model. Unfortunately, the approach yielded only a 0.01 increase in the overall recall score.

Key Takeaways

The initial testing of the Nearest Neighbor classifier marginally outperformed the results of Decision Tree classifier. We were able to augment our results by testing the Ensemble Learning technique, in particular the Random Forest and Bagging classifier; by testing hyper-parameter selection optimization, in particular the GridSearchCV technique; and finally, by testing an error minimization technique called AdaBoost technique. The highest Recall Score that we were able to obtain after testing all of these techniques was 40% for accounts with claims. The Nearest Neighbor algorithm may under perform for the same reasons noted in the key takeaway section of the Nearest Neighbor model, i.e. that the features being utilized are not sufficient to property differentiate between companies that did and did not incur claims.

III. NEAREST NEIGHBOR

In this section, we look at the Nearest Neighbor classifier. A single hyper-parameter, the number of nearest neighbors, was utilized to modify the classifier.

Nearest Neighbor – No Modifications

The initial testing of the Nearest Neighbor classifier returned subpar results in comparison to both the single Decision Tree and the Random Forest models.

Classification Report Results - Train

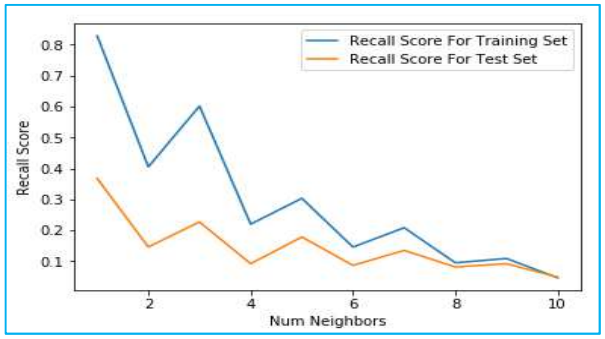
	precision	recall	f1-score	support
0	0.86	0.96	0.91	1994
1	0.67	0.31	0.42	463
avg / total	0.82	0.84	0.82	2457

Classiciation Report Results - Test

	precision	recall	f1-score	support
0	0.83	0.93	0.88	666
1	0.39	0.19	0.26	154
avg / total	0.75	0.79	0.76	820

Hyper-Parameter Section

The only hyper-parameter available for adjustment was the number of nearest neighbors. If not selected, the algorithm defaults to 5. The line graph on the right charts the change in recall score for both the training and test datasets for n-nearest neighbors ranging from 1 to 10. The highest recall score for both data sets was obtained with a nearest neighbor value of 2[10].



Enhanced Results

The second iteration of the nearest neighbor classifier performed much better once the n-nearest neighbor value was adjusted to two. The recall obtained from this iteration of the study, while below expectations, still exceeds the best score obtained from either the single Decision Tree or Random Forest Classifiers.

Classiffication Report - Train Results

	precision	recall	f1-score	support
0	0.93	0.98	0.96	1861
1	0.89	0.69	0.78	432
avg / total	0.92	0.93	0.92	2293

Key Takeaways

While the Nearest Neighbor classifier was only able to classify 40% of the test accounts with claims correctly, it did ultimately out-perform the prior two classifiers. Based on the trend in the recall score vs the number of nearest neighbors, it may be correctly supposed that there is a tight clustering and very small differences in Euclidean distance for companies that did and did not incur a claim. This may be why the recall score drops so drastically as the number of nearest neighbors increases as there are multiple very near neighbors that when taking into consideration in the vote detract from arriving at the correct prediction. Further investigation should be directed toward these distances and whether or not adjustments can be made to the existing model or if changes in the underlying features, i.e. combining features or introducing new ones would help to enhance the model.

Classiffication Report - Test Results

	precision	recall	f1-score	support
0	0.86	0.88	0.87	799
1	0.43	0.40	0.41	185
avg / total	0.78	0.79	0.78	984

[10] It is interesting to note how precipitously the recall score falls off as the number of nearest neighbors grows.

IV. NAÏVE BAYES

In this final section, we look at the Naïve Bayes classifier. While no hyper-parameters were available to adjust the model, knowing from the Data Visualization section of the study that certain features are highly correlated with claims incidence, features were sequentially dropped from the dataset to see if results could be improved.

Naïve Bayes – No Modifications

Initial results indicate that the Naïve Bayes classifiers is a more precise in terms of the Recall score generated than either the Decision Tree or the Nearest Neighbor models.

Hyper-Parameter Section

The graph to the right shows the change in the Recall score for the data set with one feature removed. The highest Recall Score results were obtained by dropped in the SIC Code, Employee Count and Revenues.

This should not be a surprise as in terms of the incidence of claims the SIC Code is highly correlated with Industry, and Employee count highly correlated with Revenues.

Final Results

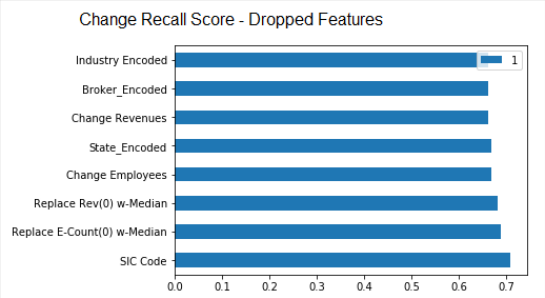
The final results yielded a recall score of 0.71 for accounts with claims and a combined 0.67 total, a full 26 bps higher than the Nearest Neighbor classifier and 30 bps higher than the Decision Tree classifier. It would appear that the Naïve Bayes approach of using conditional probabilities yielded a better result for this type of study. This may have something to do with the fact that predictions are made based on the frequency with which features are represented in the target class as opposed to a measure of Euclidean distance and or Entropy.

Training Results

	precision	recall	f1-score	support
0	1.00	0.83	0.91	1994
1	0.57	1.00	0.73	463
avg / total	0.92	0.86	0.87	2457

Test Results

	precision	recall	f1-score	support
0	0.90	0.70	0.78	666
1	0.33	0.66	0.44	154
avg / total	0.79	0.69	0.72	820



Feature Dropped => SIC Code

Training Results

	precision	recall	f1-score	support
0	1.00	0.81	0.90	1994
1	0.56	1.00	0.71	463
avg / total	0.92	0.85	0.86	2457

Test Results

	precision	recall	f1-score	support
0	0.91	0.66	0.77	666
1	0.33	0.71	0.45	154
avg / total	0.80	0.67	0.71	820

V. STUDY CONCLUSION

The objective of this study was to use Machine Learning techniques to predict companies most likely to incur an Employment Practice Liability violation. In light of the high probability that a US company will incur an EPL violation in any given year, and the potential high monetary consequences, a model that is able to predict companies most likely to incur a violation would be very useful for users and purchasers of insurance alike. The better providers and users are able to predict the likelihood of a claim the better capital allocation decision that can be made by both parties.

Phase I of the study consisted of identifying, creating and preparing our dataset for machine learning applications. This process concluded with the creation of an Analytics Based Table with 8 features and one target variable.

Phase II consisted of the analysis of our data set using measures of central tendency, frequency distribution, correlation, regression and distribution by the target feature. We observed that for each numerical feature there were a number of large outliers that could potentially skew our data, that certain features, in particular revenues and employee count were highly correlated, that our data was not normally distributed, and that certain features, in particular revenues, employee count, state and industry were highly correlated with our target variable. These observations gave us confidence that classification models would likely be the most appropriate models for our study. From the myriad of different classification models the Decision Tree, Nearest Neighbor and Naïve Bayes classifiers were chosen.

The Decision Tree classifier yielded subpar results. The model appeared to under-fit the training set which lead to subpar results in the test set. We observed in the visualization of the decision tree that certain features, include the SIC Code, Revenues and Employee count, contributed most to the Gini score, which was consistent with our findings in Phase II. A marginal improvement to the recall score was made with the application of Ensemble Learning techniques, principally the Random Forest approach, hyper-parameter optimization, principally the GridSearchCV approach, and error correction using the AdaBoostClasifier technique. We concluded that without further investigation, the Decision Tree classifier was likely not the optimal algorithm for accurately predicting our target variable from the dataset in question.

The Nearest Neighbor classifier yielded improved, yet still subpar results. The initial results of the model were inferior to that of the Decision Tree classifier in which the model was only able to predict 19% of the accounts with claims. We sought to improve the results by iterating the model over different numbers of n-neighbors. This approach yielded an optimal parameter selection of 2-3. Re-running the model with this parameter increased the model's recall score to 40% for accounts with claims. We concluded that, despite these improvements, a 40% recall score was not sufficient to consider this a successful classifier for our study.

The Naïve Bayes classifier yielded the best results of all three models. The initial results yielded a 67% recall score. While the Naïve Bayes classifier did not provide for hyper-parameter selection, we hypothesizes that by sequentially excluding certain features that were highly correlated we could obtain a higher recall score. By implementing this modification, we were able to increase our recall score for accounts with claims to 71%, far superior to the result obtain from either the Decision Tree or Nearest Neighbor models.

While the Naïve Bayes model yielded the best results further time should be dedicated to understanding why the Decision Tree and Nearest Neighbor classifiers generates poor results. These results may have something to do with the feature variables of our study, in which the models were unable to identify sufficient dissimilarities between companies with and without claims to make an accurate prediction. Ultimately, a more robust dataset with more features should be create to retest these models.