Process Mining to aid in the Claims Litigation Predictive Modeling Process

Author: Corey Arnouts

Abstract

Usually, one thinks of healthcare costs as being covered by health insurance companies, however, In Michigan, Personal Injury Protection has been a unique auto insurance coverage because auto insurers have paid for almost all of the costs of injuries associated with an auto accident. Depending on an insured's healthcare plan, auto insurers may have to act as either the Primary or Excess insurer for medical costs related to auto accidents. Michigan, up until recently, was also no-fault state and had unlimited limits on this Personal Injury Protection(PIP) Insurance. This meant that no matter how high the cost of a Claim the medical coverage did not stop. This is an imperfect system that is slowly changing. This system has also caused an environment of litigation to occur between insurance companies and medical providers because medical providers have had the ability to charge as much as possible on injuries that occurred during automobile accidents because they know the insured or patient has unlimited coverage and that their auto insurer will cover whatever they charge. For example, a MRI for a non-auto-related injury may cost $1,000, while the same MRI for an auto-related injury may cost as much as $10,000 or more. Medical Providers have done this because it does not adversely affect their clients; they are still receiving the treatment they need. The auto insurance companies are footing the bill and ultimately increasing auto rates to compensate for these high costs. Astronomical costs have caused PIP Claims to be heavily litigated in Michigan, as insurers refuse to pay unreasonable prices and providers are subsequently suing the insurers. A high percentage of Farm Bureau of Michigan PIP claims have some kind of litigation occurring on them, somewhere in the neighborhood of 10%, litigated claims are increasingly expensive because insurers have to pay legal costs and may still be stuck with the full cost of the medical bills, because of this insurers would like to avoid PIP Litigation when possible, and when litigation is inevitable they would like to not pay for medical services that are overpriced. To do this, insurers would like to identify which PIP Claims have a high likelihood of litigation so they can develop a claim-handling strategy and take actions early in the Claims lifecycle to prevent litigation and adverse development of claim costs.

Insurers such as Farm Bureau of Michigan have taken actions to reduce costs and inspect suspicious claims early in the Claims lifecycle. Still, many of the efforts have not been as coordinated and as data-driven as they could be. Through the use of analytical techniques, the hope is that new methodologies can be developed for handling Personal Injury Protection Claims. These will involve the implementation of predictive models and also an overhaul of the Business Processes dealing with these Claims.

In order to overhaul Business processes and understand claim life cycles, a field known as Process Mining is going to be used, Process Mining uses event logs from Information Systems to look at how Processes play out from a data perspective. Using this event log data, organizations can discover how their processes are being executed and work to improve further and optimize these processes.

Predictive Modeling is also a field in recent years that has gained a lot of attention. Predictive Modeling uses data from the past to predict what will happen in the future; it pairs data with statistical models in order to do this. Predictive Modeling has a rich history in the Auto Insurance space; using loss forecasting to predict vehicle damages has been a part of the industry for a long time. To a large degree, Predictive Modeling in the Insurance space is based on static modeling elements such as the loss history of an Insured, the age of an Insured, the type of vehicle, the size of the home, and the Insurance Score, which is a metric developed off of credit reporting elements in order to predict future losses. Process-

based feature engineering will bring more dynamic and informative data to the Insurance sector, making decisions not just based on static elements like the age of the insured and location but also on the actual action that the insured or customer is taking. Process-based feature engineering will allow insurance companies to understand how the clients interact with the goods and services.

Key Words

Process Mining – The discipline of extracting insights from process-oriented event log data

Process Trace – Specific path that a process can follow

Process Instance – a single execution or iteration of a particular process

Predictive Modeling/Machine Learning – Algorithms that use data about the past to make predictions about the future.

Claims Litigation - Lawsuits involving Insurance claims, often involving the insurer, insured, and third parties such as medical providers

Personal Injury Protection – a component of automobile insurance that covers the healthcare expenses associated with a car accident.

Provider Network Metrics – Statistically derived metrics around the proclivity of certain medical providers to end up in legal battles around auto claims they are involved in.

Household Metrics – Measures of profitability of customers can be thought of as a more holistic view of a customer, as they bring in data from different lines of business and different members of a family or Household.

Gradient Boosting Machine – a machine learning method that uses tree-based weak learners to make predictions and then measure the error function and use subsequent models to reduce the error. Generating an increasingly accurate algorithm.

Problem Statement

To create a Predictive Model that predicts with a high degree of accuracy whether a Personal Injury Protection claim will end up in litigation. I am making the prediction based on the information that is available on the Claim within 60 days of the Claim being reported or before the Claim the litigation, whatever length of time is shorter. I am building the training set off of claims that occurred in 2016, 2017, and 2018 and have a Personal Injury Protection exposure on them. I will then test the predictive model on the first 6 months of 2019; the reason that the testing data is from 2019, not 2020, is because PIP Claims play out over a long period of time, so in order to truly know whether or not a claim will end up in litigation, the Claim needs a at least year to go through it's lifecycle. In order to accomplish this goal I plan on using predictive features like Claim Loss Location, Medical Providers that are parties on the Claim, along with features about the client themselves. I engineer predictive features that are not only based on a single point in time but rather are also based on the interactions we have had with the client in the past; this will lead to a more dynamic and informative view of our customers. Using all of this data together, I can make accurate predictions on the future outcomes of PIP Claims.

Objective

To prove that whether or not an Auto Insurance Claim ends up in litigation is predictable using data elements gathered during the Claims Lifecycle, and that Process Mining can be used on Complex Event Logs to help engineer both Predictive features and target variables to be used in Predictive Modeling Initiatives, that will ultimately increase the profitability of the company.

Related Work (Literature Review)

A lot of work exists in both the spaces of Machine Learning and Process Mining, but there is significantly less research on the convergence of the two. Process Mining was started by Wil Van Der Aalst as a way to understand and visualize processes with the use of data[1]. This is viewed as a separate discipline from AI and Machine Learning, as they are both distinct fields of study that both depend heavily on data and are both benefiting tremendously from the exponential increase of data availability. There has been some work that looks to leverage both Process Mining and Machine Learning together to achieve the desired outcome. The work presented in [2] uses Discrete event transition probabilities within event logs to predict specific process outcomes. This work discusses how there can be difficulties predicting process outcomes when the Process is currently running, but this is the only valuable type of model to build. The work also discusses particular possibilities like seasonal drift that can occur in Process-based Predictive Models. To make predictions, the models that are built not only use previous Process traces that look exactly like the process instance being predicted. They also use Jaccard Similarity and Damerau Levenshtein distance as a way to identify similar Process Instances. This is a really creative way of identifying processes that are similar but are not precisely the same. The different process traces and transitions are engineered into features using one hot encoding. This paper makes predictions on the remaining amount of time in the Process Instance and also the future path of the Process Instance. They call the prediction method a Similarity-based Transition system. In work presented in this paper I am heavily focused on the future outcomes of the process instances, with each Claim being an instance, and the outcomes I am focusing on are PIP litigation. In other work [13] shows how more advanced modeling techniques can be applied to predict process outcomes, [13] shows how the use of Neural Networks can be successful in process prediction. [3] Shows an approach that uses the Instance-specific Probabilistic Process Models (PPM) and is able to show that similarly to [2] event-based semi-structured business processes can be successfully predicted. The work in [3] is very similar to the PIP Claims Processes that I am looking to predict because these processes I would consider to be semi-structured as well. The research in [6] provides a framework for correlating target variables with process instances to find underlying data patterns that can be leveraged. The work in [4] and [9] show that through the use of Process Mining and Machine Learning, we can figure out how to optimize our resources best to handle Business Processes; this is also very relevant in the use cases that I am exploring because ultimately the output of the model will be used to better allocate human resources to facilitate and execute the processes in a way that will ultimately reduce costs. At the end of the paper, I will present some of the tools that I am hoping to use to better monitor process instances.

Research presented in [7] and [8] propose ways in which Network Based metrics can improve Predictive Models. They show how social network analysis can highlight irregular behaviors that may be helpful for predicting specific target variables; I utilize a similar methodology when I am coming up with medical provider litigation metrics that are indicative of the proclivity of certain providers to end up in Litigation circumstances. The providers that are involved in a particular claim can be very useful in determining whether or not the Claim will be litigated, because, ultimately the act of litigation is determined by either the Provider or the insurer.

[5] and [10] highlight some of the existing work that has been done with both process mining and machine learning in the Auto Insurance space; the development of these techniques in the insurance sector is early on in its maturity cycle but is gaining momentum quickly.

Methodology

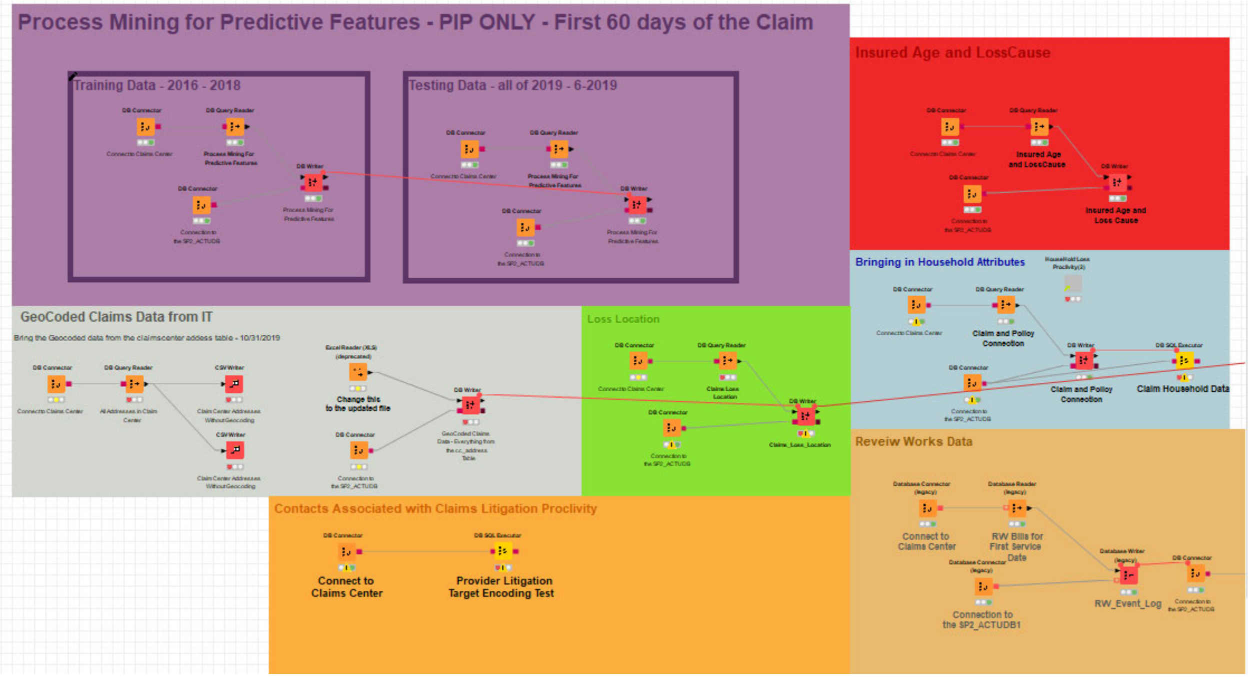
Data Gathering

For my data gathering, I used data from of a number of different Enterprise Systems and applications at Farm Bureau Insurance of Michigan. We use Guidewire as our primary administration system, and this comes with multiple components. The primary focus of this analysis was on Guidewire ClaimCenter data; this admin system houses all of the relevant information for every Farm Bureau insurance claim, from the start of the claim lifecycle to the end.

The Household-based Metrics I developed and further discuss in the section below are pulled from our CRM (Customer Relationship Management) system, FBCares. The advantage of using features from a CRM is the fact that it contains a mix of both legacy system data and data from updated systems. So it is a good place to get a holistic view of a customer and the Household they belong to. The FBCares CRM system also allows us to view all lines of business at the same time. This system will enable me to get a better understanding of the client.

Extract Transform and Loading Data

During the Data Gathering phases I had to move a lot of data from disparate sources onto the same server to merge, blend, and enrich the data. In order to move data from multiple sources, I utilized an open source Data Science Platform known as Knime. Knime allows me to execute and an entire workflow of the data science process at once. I spent quite a bit of time connecting Knime to our SQL Servers at Farm Bureau, but it has proved very beneficial. I can combine SQL Scripts, R Scripts, and Python Scrips all in one cohesive workflow, and Knime also has a lot of Data Science Functionality of its own that is accessible.



Process Mining for Predictive Features - PIP ONLY - First 60 days of the Claim
Training Data - 2016 - 2018
Testing Data - all of 2019 - 6-2019
DB Connector
DB Query Reader
DB Writer
Connectto Claims Center
Process Mining For Predictive Features
Connection to the SP2_ACTUDB
Insured Age and LossCause
Insured Age and LossCause
Insured Age and Loss Cause
Bringing in Household Attributes
HouseHold Loss Proclivity(3)
Claim and Policy Connection
DB SQL Executor
Claim Household Data
GeoCoded Claims Data from IT
Bring the Geocoded data from the claimscenter addess table - 10/31/2019
CSV Writer
All Addresses in Claim Center
Claim Center Addresses WithoutGeocoding
Excel Reader (XLS) (deprecated)
Change this to the updated file
GeoCoded Claims Data - Everything from the cc_address Table
Loss Location
Claims Loss Location
Claims_Loss_Location
Reveiw Works Data
Database Connector (legacy)
Database Reader (legacy)
Database Writer (legacy)
Connect to Claims Center
RW Bills for First Service Date
Connection to the SP2_ACTUDB1
RW_Event_Log
Contacts Associated with Claims Litigation Proclivity
Connect to Claims Center
Provider Litigation Target Encoding Test

Fig1 – Picture of the full data science workflow built in knime that orchestrates all of the ETL processes and also Executes the R script for the predictive model, all of the nodes are either connecting to SQL databases, executing SQL code or reading and writing data

Challenges with Data Gathering and Aggregation

There are some difficulties, however when combining data across multiple systems, for example, often policy numbers are listed differently across different systems, and data for the same Claim can be a mix of legacy and new data. To remedy this, I have to perform a number of different manipulation techniques to get the data to match. I created a user-defined function in SQL that takes just the numeric characters from a string, this has proven to be useful, but other manipulations have to be done as well. Sometimes a policy number in a particular system may have a prefix in front of it but will not have a prefix in another system.

Process Oriented Data

There can also be challenges inherent to gathering and utilizing Process-oriented data, in the section below, I discuss the various elements required for process mining, but these data points do not exist in that format in their raw forms. There is a process by which the data has to manipulated so that it fits within the event log data structure. The case identifier is the piece of data that ties the Process together with insurance claims fortunately, the case identifier is relatively straightforward because every Claim is assigned a number. The other elements can be more ambiguous, though; the activity column can be anything that you want it to represent, whether it is an activity that an internal employee takes or an event that an external participant engages in, or it may be something that just occurs without any intention like the accident that causes the Claim. The timestamps around process data can also be a challenge at times, there is not always an end timestamp to every event, so you have to understand the context of the Process and what you are trying to measure. The claim loss date is a good example; there is really only one distinct time associated with it, so in a case like this, I would decide to use this time for both the start and end timestamps.

Extract the Data

The first step is to find all of the needed data from across the organization; this data may be found in different systems and will give different perspectives on how other Business Processes and Client Interactions are playing out. This data will include data that is generated from our employees, our agents, our clients, and our systems. All of the data sources used will need to contain timestamps that so they can be assembled in a chronological order that can then be mined for patterns. Even non Process Related features like the location of the Loss will need timestamps so that we understand the point in time that they occur.

Transform the Data

I will need to put the data in a format in which it can be combined with some of the key aspects of Process Mining data are:

Case ID: Represents the Party that is going through the Process (Person, Policy, Claim, etc..)

Activity/Event: The event that has taken place

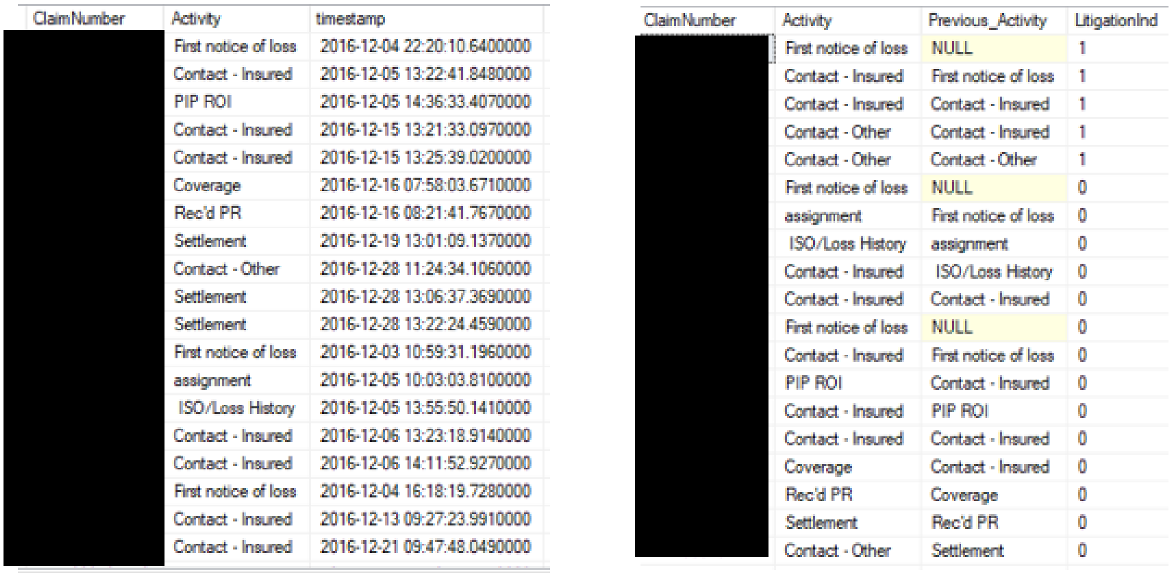
Activity Instance ID: Unique Identifier for the Activity

Resource: Person or system responsible for executing the event

Start Datetime: start of the event

End Datetime: End of the event

All of these data components are necessary to construct a process mining log. Once the process data is assembled I match the process data with the relevant target variables that I am looking to predict. So that I can then mine patterns that correlate the event log with the target variable that I am looking to predict.



ClaimNumber	Activity	timestamp
[redacted]	First notice of loss	2016-12-04 22:20:10.6400000
[redacted]	Contact - Insured	2016-12-05 13:22:41.8480000
[redacted]	PIP ROI	2016-12-05 14:36:33.4070000
[redacted]	Contact - Insured	2016-12-15 13:21:33.0970000
[redacted]	Contact - Insured	2016-12-15 13:25:39.0200000
[redacted]	Coverage	2016-12-16 07:58:03.6710000
[redacted]	Rec'd PR	2016-12-16 08:21:41.7670000
[redacted]	Settlement	2016-12-19 13:01:09.1370000
[redacted]	Contact - Other	2016-12-28 11:24:34.1060000
[redacted]	Settlement	2016-12-28 13:06:37.3690000
[redacted]	Settlement	2016-12-28 13:22:24.4590000
[redacted]	First notice of loss	2016-12-03 10:59:31.1960000
[redacted]	assignment	2016-12-05 10:03:03.8100000
[redacted]	ISO/Loss History	2016-12-05 13:55:50.1410000
[redacted]	Contact - Insured	2016-12-06 13:23:18.9140000
[redacted]	Contact - Insured	2016-12-06 14:11:52.9270000
[redacted]	First notice of loss	2016-12-04 16:18:19.7280000
[redacted]	Contact - Insured	2016-12-13 09:27:23.9910000
[redacted]	Contact - Insured	2016-12-21 09:47:48.0490000

ClaimNumber	Activity	Previous_Activity	LitigationInd
[redacted]	First notice of loss	NULL	1
[redacted]	Contact - Insured	First notice of loss	1
[redacted]	Contact - Insured	Contact - Insured	1
[redacted]	Contact - Other	Contact - Insured	1
[redacted]	Contact - Other	Contact - Other	1
[redacted]	First notice of loss	NULL	0
[redacted]	assignment	First notice of loss	0
[redacted]	ISO/Loss History	assignment	0
[redacted]	Contact - Insured	ISO/Loss History	0
[redacted]	Contact - Insured	Contact - Insured	0
[redacted]	First notice of loss	NULL	0
[redacted]	Contact - Insured	First notice of loss	0
[redacted]	PIP ROI	Contact - Insured	0
[redacted]	Contact - Insured	PIP ROI	0
[redacted]	Contact - Insured	Contact - Insured	0
[redacted]	Coverage	Contact - Insured	0
[redacted]	Rec'd PR	Coverage	0
[redacted]	Settlement	Rec'd PR	0
[redacted]	Contact - Other	Settlement	0

Fig2 – Data tables that were a part of the intermediary steps to get the process transition matrix that generated the process related features

In the pictures above, you can see how I assemble the data in chronological order and then use the lag function in SQL to assemble the data so that the cases transition from one event to another. The transitions are then aggregated from this point to see the average litigation indicator for each activity transition. This idea is further explored in the section below.

Feature Engineering

Feature Engineering is perhaps the most critical aspect of the Machine Learning process and it is where the creativity of person performing the analysis is required. One of the most important considerations when doing predictive modeling initiatives is the timing of when certain variables will be available for use. We want to train predictive models that will be able to use the data that is available at the time they are making live predictions. For this predictive modeling initiative, I built a model that would be implemented at the 60-day mark of the Claims Lifecycle; this is relatively early in the PIP Claim Lifecycle as many PIP claims go on for many months or even years. 60 days from the start of the Claim also provides us with plenty of information about the progress of the Claim, such that we can make accurate predictions about the eventual outcome of the Claim. With that being said, all of these variables are engineered in a way such that they represent information encoding that would be available about the Claim at this 60-day mark.

Process Oriented Metrics (Target Mean Encoding Discrete Event Transitions)

A common practice in Machine Learning is to encode categorical variables with the mean of whatever we are trying to predict; in my case I am trying to predict whether or not a certain claim will end up in litigation, during the Claims Process there is a lot of different steps that take place and in our admin system, most of these events are indicated. In the Claims Department, there is an information system called ClaimsCenter that handles every aspect of the Claim, from the Claim being reported to the Claim being closed. This information system stores data about the Claim itself, how it was reported, the actions taken on it, the transactions involved with the Claim, etc…. Within this system there are Events that are logged on the Claim depending on what happens to the Claim, some of these activities are displayed below.

All of these activities occur during the onset of the Claim (within the first couple months); these activities are only a small subset of all of the possible events that occur on the Claim during this time. I paired the Process Event Log Data of historical Claims with a Boolean target variable (0,1) representing whether or not that Claim eventually went to litigation and looked at the percent of Claims that had certain sequences of activities and went to litigation. That is what is represented in the Graphic above. The number within the Activity bubbles represents the number of times that activity occurred on PIP Claims, and the number on the transitions between activities represents the percentage of Claims that took that path that eventually went into litigation. For example, a Claim that has a PIP Activity occur before the Contact-Insured Activity ends up in Litigation 17% of the time.

Coming up with the Process Based Metric (Simplified diagram)

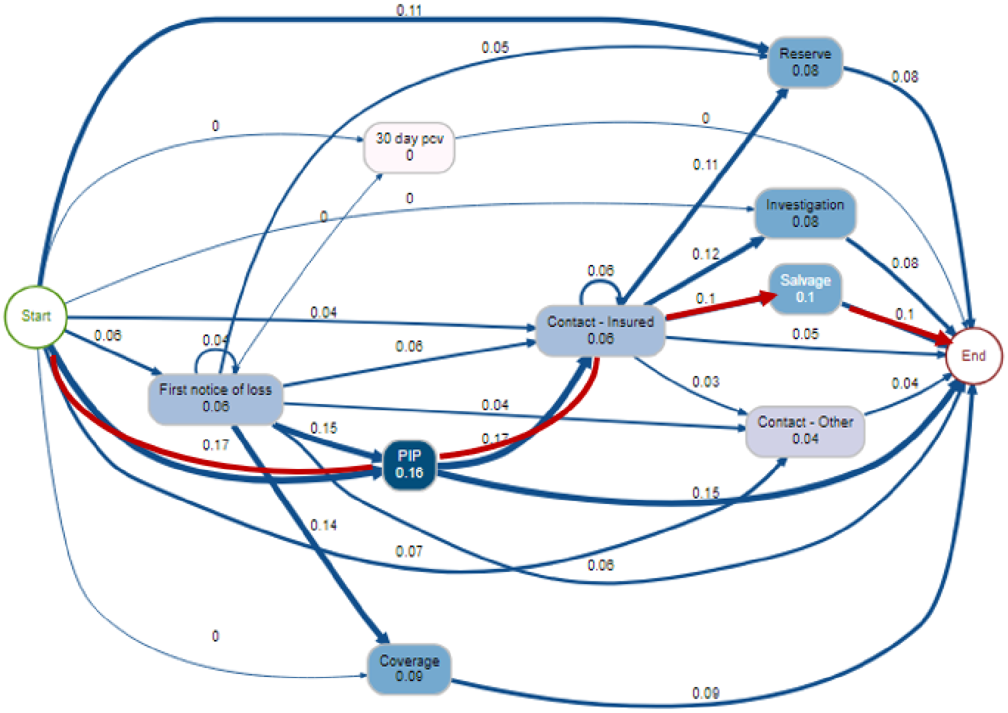
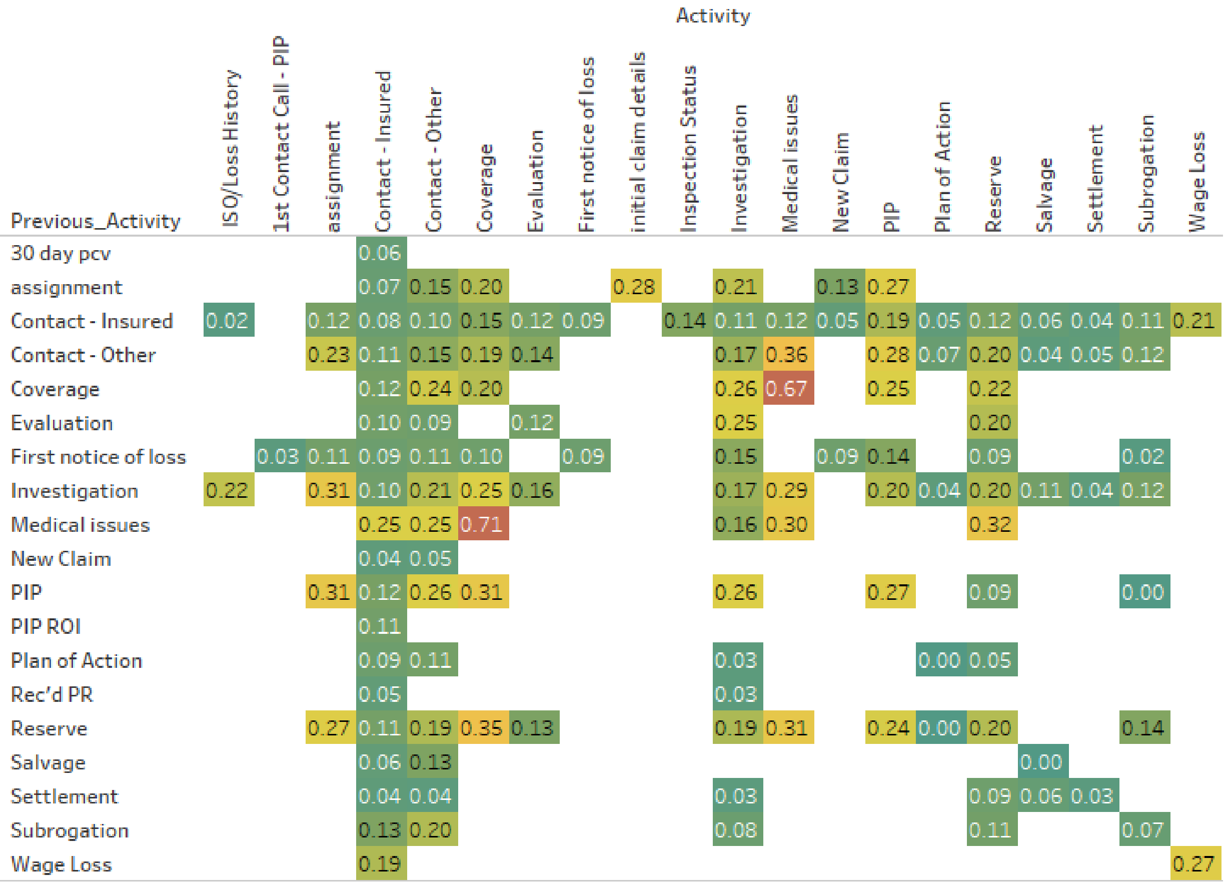


Fig3 – The same diagram as figure 1 except a specific path is highlight. The Path above would be represented by the following equation:(.17+.17+.1+.1)/4 = .135, for Claims that followed this specific path their Process based Litigation Encoding would be .135. This roughly equates to saying “based on this Claim’s path only it has a 13.5% chance of going into Litigation”

Engineering Process-Based Features

Essentially I am using the Activities that occurred at the Beginning of the Claim to understand the likelihood that the Claim will eventually end up in litigation. I did this on all the Activities Event Log data that occurred on the Claim during the first two months of the Claims Lifespan or before the Claim ever went to litigation, whatever period was shorter. I then came up with an aggregate Metric called Process

Based Litigation Proclivity, which was the average of all the Transition Likelihoods. This variable was one of the features I used in the Litigation Predictive Model. I created a training set of data based on 2017 – 2019 and my test set based on the first half of 2019 data. The Activity Transition Litigation Likelihoods were mapped to the test set to come up with the aggregate Process Based Litigation Proclivity Metric of these Claims.



	Activity																			
Previous_Activity	ISO/Loss History	1st Contact Call - PIP	assignment	Contact - Insured	Contact - Other	Coverage	Evaluation	First notice of loss	initial claim details	Inspection Status	Investigation	Medical issues	New Claim	PIP	Plan of Action	Reserve	Salvage	Settlement	Subrogation	Wage Loss
30 day pcv				0.06																
assignment				0.07	0.15	0.20			0.28		0.21		0.13	0.27						
Contact - Insured	0.02		0.12	0.08	0.10	0.15	0.12	0.09		0.14	0.11	0.12	0.05	0.19	0.05	0.12	0.06	0.04	0.11	0.21
Contact - Other			0.23	0.11	0.15	0.19	0.14				0.17	0.36		0.28	0.07	0.20	0.04	0.05	0.12	
Coverage				0.12	0.24	0.20					0.26	0.67		0.25		0.22				
Evaluation				0.10	0.09		0.12				0.25					0.20				
First notice of loss		0.03	0.11	0.09	0.11	0.10		0.09			0.15		0.09	0.14		0.09			0.02	
Investigation	0.22		0.31	0.10	0.21	0.25	0.16				0.17	0.29		0.20	0.04	0.20	0.11	0.04	0.12	
Medical issues				0.25	0.25	0.71					0.16	0.30				0.32				
New Claim				0.04	0.05															
PIP			0.31	0.12	0.26	0.31					0.26			0.27		0.09			0.00	
PIP ROI				0.11																
Plan of Action				0.09	0.11						0.03				0.00	0.05				
Rec'd PR				0.05							0.03									
Reserve			0.27	0.11	0.19	0.35	0.13				0.19	0.31		0.24	0.00	0.20			0.14	
Salvage				0.06	0.13												0.00			
Settlement				0.04	0.04						0.03					0.09	0.06	0.03		
Subrogation				0.13	0.20						0.08					0.11			0.07	
Wage Loss				0.19																0.27

Fig4 – This Diagram shows the various Litigation proclivities for the various activity transitions, the activity on the left axis is the previous activity and the activity on the x axis is the activity that the claim is transitioning to. The number in the box represents the litigation percent of claims that travelled that specific path

The above matrix shows a majority of the transition matrix that is used to determine the Process Based Litigation Proclivity; as you can see, there are certain patterns that reveal themselves to be predictive, for example, having a coverage activity and then a medical Issues activity is indicative of a Claim that will go to litigation as 67% of Claims that have taken this path have gone to litigation in the past. While other paths are much more innocuous, such as moving from New Claim to Contact-Insured, this is a pretty common and straightforward path, a Claim happens, and then we have some Contact with the Insured to talk about what happened. This path only has a litigation percentage of 4%, meaning claims that follow this path rarely end up going to litigation.

Household Metrics

There is often a desire for companies to be able to have a 360-degree view of a customer; knowing as much about customers as possible is a great way to make informed business decisions; Household based metrics are an attempt at doing exactly this. Household Metrics are metrics that are based off of all the business that we have with a specific customer so if they are associated with 10 policies on our book then data from all ten policies will be included. It is somewhat similar to Customer Lifetime Value metrics used in marketing. The Metrics focus on how profitable the Household has been when they have been a client. Households consist of several different people who are then connected to a number of policies; some of these people may be on the same policy, but others may be on different policies, but at one time, they were on the same policy. Household rules are not necessarily strictly defined at Farm Bureau, but represent connections between people and the policies they have or have had in the past. Looking at metrics on this level further lets us understand clients on a more holistic basis, and client profitability in one line of business has been shown to predict profitability in other lines of business as well.

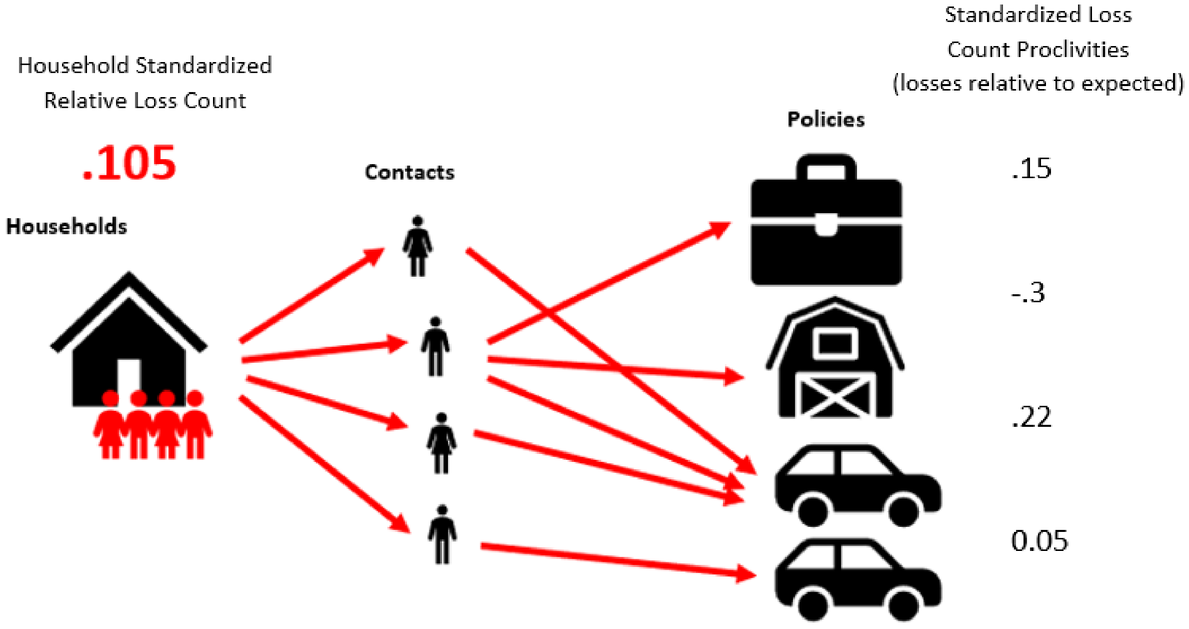


Fig5 – This Diagram shows a visual representation of how the household metrics are created and how they span across multiple insureds and multiple policies

HHStandardizedRelativeLossCount = This metric is derived by looking at all of the policies that the specific Household is apart of, and the amount of time that those said policies have been open. This metric is meant to compare how many losses this Household has had vs. how many losses we would expect them to have based on the amount of time they have been a customer of ours. Then we calculate essentially a Z-Score for the specific policy based on the number of Claims a policy has had vs. the number of claims we expected the policy to have. So let's say on average, an auto policy has .1 claims per year with a standard deviation .9. Over a 10-year period, we see that a specific policy has 3 Auto Claims, so to calculate the Z score of this policy we do the following:

z = (x-µ)/σ), Z= (3 –1)/9, therefore, the "Policy Standardized Relative Loss Count" would be 2/9 or .22, which would mean that this policy has a relatively high loss proclivity. Anything over zero is perceived as an above-average Loss Proclivity.

We then take the Standardized score for every policy that the Household has across all lines of business and then aggregate these using an arithmetic average. So if a Household has a Home, Auto, and Farm Policy, then we would take the standard of the Standardized Loss Counts across all of these policies. This final average metric is the HHStandardizedRelativeLossCount which essentially represents the loss proclivity of an entire household from a frequency perspective.

The reason that the standardization is important is that when you are aggregating across multiple lines of business, it is important to understand that different lines have different loss proclivities and different loss distributions, so in order to truly calculate a fair metric, we have to aggregate Z-Scores as opposed to just dividing by the average. The Z-score allows us to account for some lines of business having more variation in their loss frequency distributions.

HHStandardizedRelativePremium = Same methodology as above, except this, features attempts to understand the amount of premium that we are collecting from a client relative to what one would expect based solely on the policies they have.

HHStandardizedRelativeLossAmount = Same methodology as above, but this feature is looking at the loss amount of the Claims, so it is comparting the expected amount of losses for each policy to the actual amount of losses for each policy taking the Z-score based off of this and then aggregating the metric across all lines of business

HHStandardizedRelativeLossRatio = Loss Ratio is a common term that is used in Insurance; it is often used as an Accounting Figure for an entire company or Line of Business it is essentially: (Losses/Premium Collected), so how much you had to pay for the exposures; that you cover vs the amount of premium that you collected to cover those exposures, it is a good indicator of how your company is doing. For this Household Metric, I did a similar thing and calculated the Loss Ratios for each Policy and Line of Business and then compared these loss ratios for the expected loss ratio based on the policy type and then aggregated these measures up to a Household Level.

So are these Household Features Predictive?

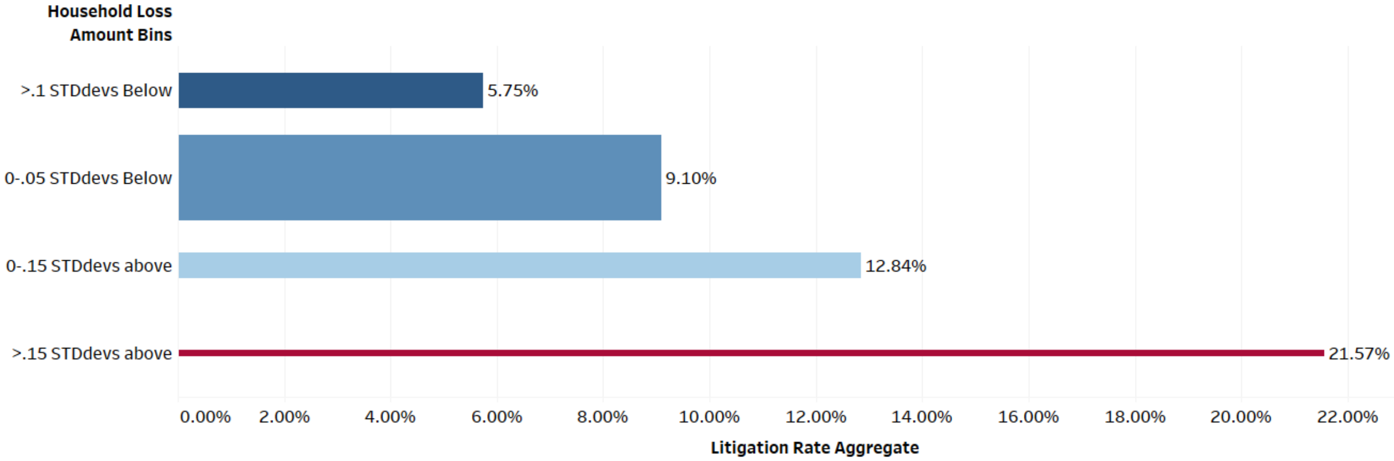


Fig6 – Litigation rates according the household loss amount classification that the policy/insured falls into, the color of the bar represents Litigation percent and the width of the bar represents policy count

As you can see from the Graphic above, the Household Relative Loss Amount Feature is an excellent way to predict the litigation proclivity of a specific Claim. Higher Household relative loss amounts correlate well with litigation proclivities, as you see from the graph above. Any claim that belongs to a household with a relative loss amount more than .15 standard deviations above the average has almost double the likelihood of having a PIP Claim Litigated. This shows how historical data on households predicts client behavior in the future.

Brief Explanation of Data Leakage

Data Leakage is a phenomenon when the predictive modeling features contain some sort of encoding of the target variable. When making predictions, you only want to be using information that would be available at the time of the prediction. In this paper, different forms of target variable encoding are being done in some of the features, including the process mining features. Any form of target encoding can lead to some degree of data leakage in the training set. I have been sure though that there is no data leakage occurring in the test set. The test set is built off of claims with reported dates in the first 6 months of 2019. All of the target mean encoding that is occurring is built off of claims that occurred before the start of 2019. So even the target mean encoding that is present in the testing data, is mean encoding that was built from data in the training set.

Claims Contact and HealthCare Provider Metrics

In the current Personal Injury Protection (PIP) Insurance environment, there are Medical Providers that try to take advantage of the fact that Insurance Companies have to pay the full bill for medical treatments related to car accidents, and in till recently this coverage was unlimited for everyone. In order to capture information about which providers are most likely to end up in litigation with Farm Bureau, I created metrics that measure the proclivity of a Medical Providers' propensity for litigation. In order to do this, I connected all lawsuits and legal matters to their relevant Claims. I then came up with a metric that compared the number of total PIP Claims that a Provider is associated with to the number of Lawsuits that said Provider is involved with. The number of Lawsuits that Provider is involved with is then divided by the total number of claims the Provider is on. This is equal to the Provider Lawsuit proclivity. The next step was to aggregate all of the Provider Lawsuit Proclivities on each Claim for Providers that were on the Claim within the first 60 days of the Claim. So the average of the Provider Lawsuit Proclivity for the entire Claim then became the "Provider Lawsuit Average" feature that you see in the modeling data. This metric is derived off of Claims between the start of 2016 and the end of 2018, so it is built off of roughly three years of data. The training data set for this modeling initiative was also built of 2016 – 2018 and the testing set was built off of the first half of 2019. So doing this Provider Lawsuit Average variable meant that there was going to be some data leakage into the training dataset because the Lawsuits that we are trying to predict are also encoded somewhere in the Provider Lawsuit Average variable. This could potentially lead to some overfitting of the data. The testing dataset will not have data leakage problems because all of the claims in the testing set happened after 2019 and the variable is built off of claims that occurred before that.

Provider Lawsuit Average Metric diagram

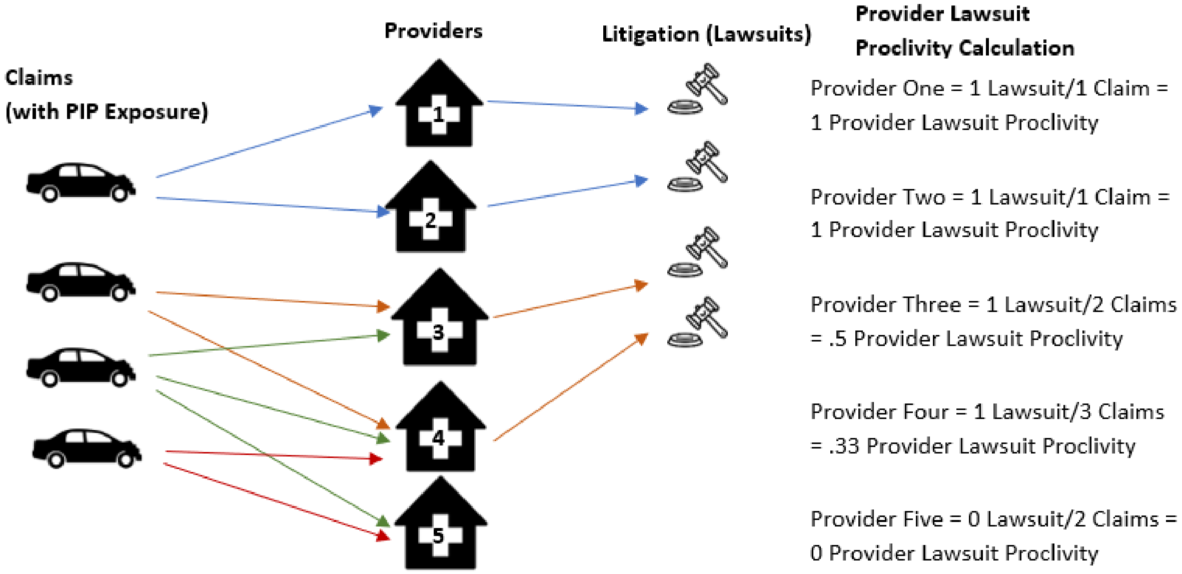

Claims
(with PIP Exposure)
Providers
Litigation (Lawsuits)
Provider Lawsuit
Proclivity Calculation
1
2
3
4
5
Provider One = 1 Lawsuit/1 Claim =
1 Provider Lawsuit Proclivity
Provider Two = 1 Lawsuit/1 Claim =
1 Provider Lawsuit Proclivity
Provider Three = 1 Lawsuit/2 Claims
= .5 Provider Lawsuit Proclivity
Provider Four = 1 Lawsuit/3 Claims
= .33 Provider Lawsuit Proclivity
Provider Five = 0 Lawsuit/2 Claims =
0 Provider Lawsuit Proclivity

Fig7 – This diagram explains how the provider lawsuit proclivity metrics are calculated on a by provider basis

This diagram above shows the relation between auto claims Providers and Litigation and how the Provider Lawsuit Proclivity Calculation is calculated on the Provider Level. If we assign a Provider Lawsuit Average metric for a Claim that says Providers 1,3, and 5. Then the Provider Lawsuit Average for that Claim would be (1+.33+.5) = .61.

Predictive Power of Provider Lawsuit Proclivity Average

On the chart below, the left-hand axis represents groupings of the Provider Lawsuit Proclivity Average. As you can see, while the lower group of the metric has a majority of the claims, the higher sets of the metric have much higher lawsuit Proclivities. The data below is built off of the testing data so there is absolutely no data leakage in the data below. Basically, proving that providers with litigation in the past, will likely have more litigation in the future

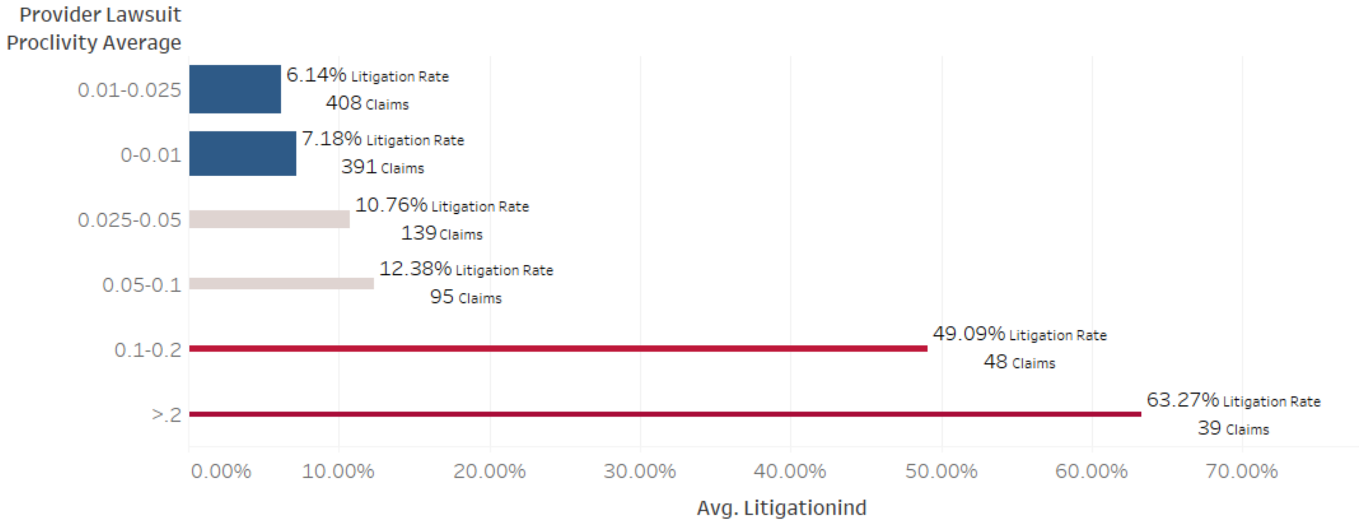


Fig8 – This diagram shows the predictive power of the Provider Lawsuit Proclivity Average

Medical Review Data

Often bills received by insurers are reviewed by medical billing companies to see if the bills are legitimate. This medical billing data can also be used for the predictive litigation model. Looking at whether or not the Claim had medical bills that were sent for bill Review within the first 60 days of the Claim can give us some helpful insight into whether or not the Claim will end up in litigation because most of the litigation is around the charges on said Medical Bills. In the graph below, we focus on one of these billing attributes: when does the insured receive their first medical service after the accident.

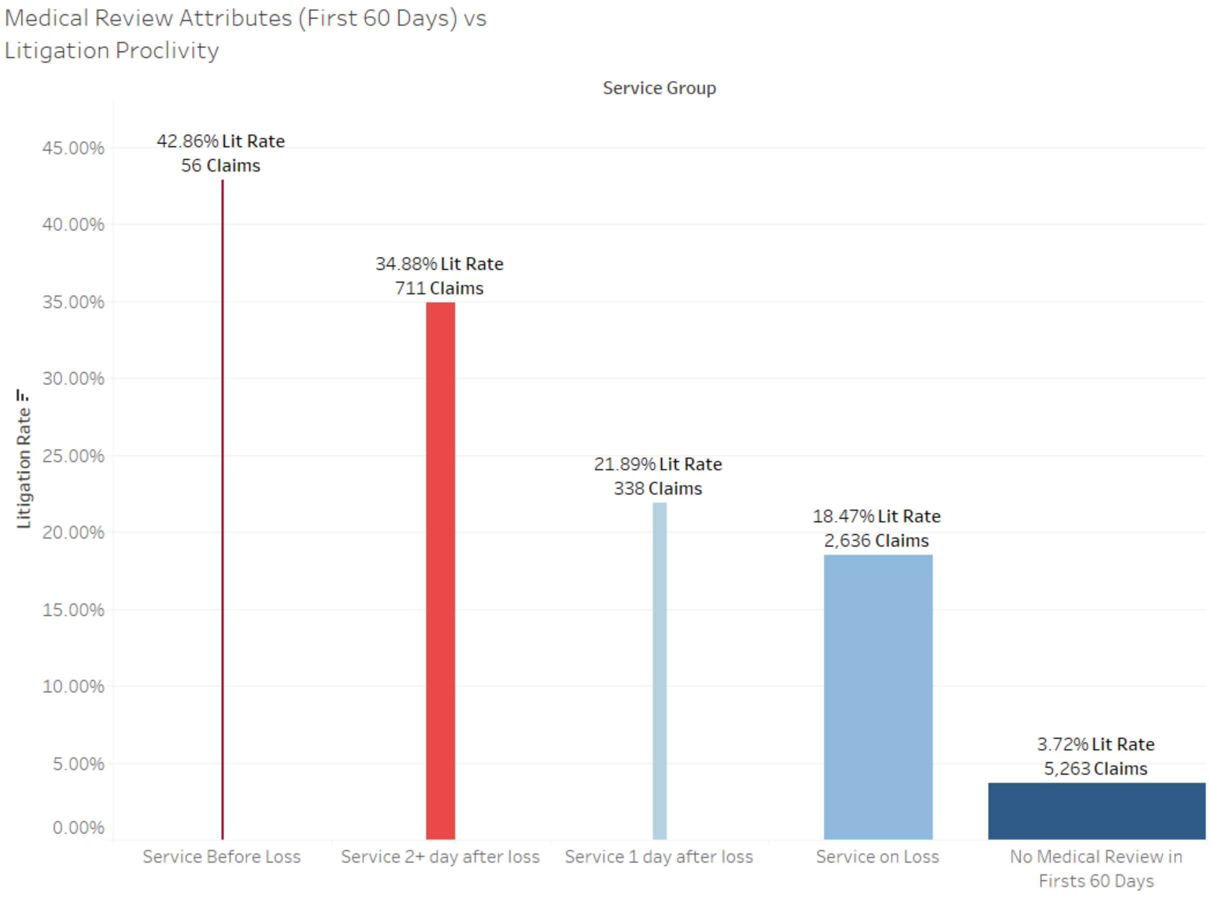


Fig9 – This diagram shows the predictive power of the First Service Date

The color in the chart represents the percent of Claims that go into Litigation for each Service Group; the different Service groups are differentiated by looking at when the insured got their first treatment relative to the loss date. As you can see, there is even a subset of Claims that had a Medical service before the loss date; the only explanations for this are either bad data or fraudulent behavior because it is impossible to get treatment for an injury you haven't had yet. The next most litigated category claims that had their first service date two or more days after the loss. This may also be suspicious because if you get injured in a car accident, you would expect to get some medical attention either that day or the next day. This variable is a good indicator of claims litigation because it shows us what Claims we should be more suspicious of according to when the injured parties got medical attention. The expectation is that legitimate PIP claims would get medical attention the same day that the accident occurred. This is also a good feature because it is available early in the claims lifecycle. Also included in the medical review data are features that indicate the number of times that someone had received medical attention in the first 60 days of their Claim, and the number of medical bills that were received in the first 60 days of the Claim.

Geospatial Analysis

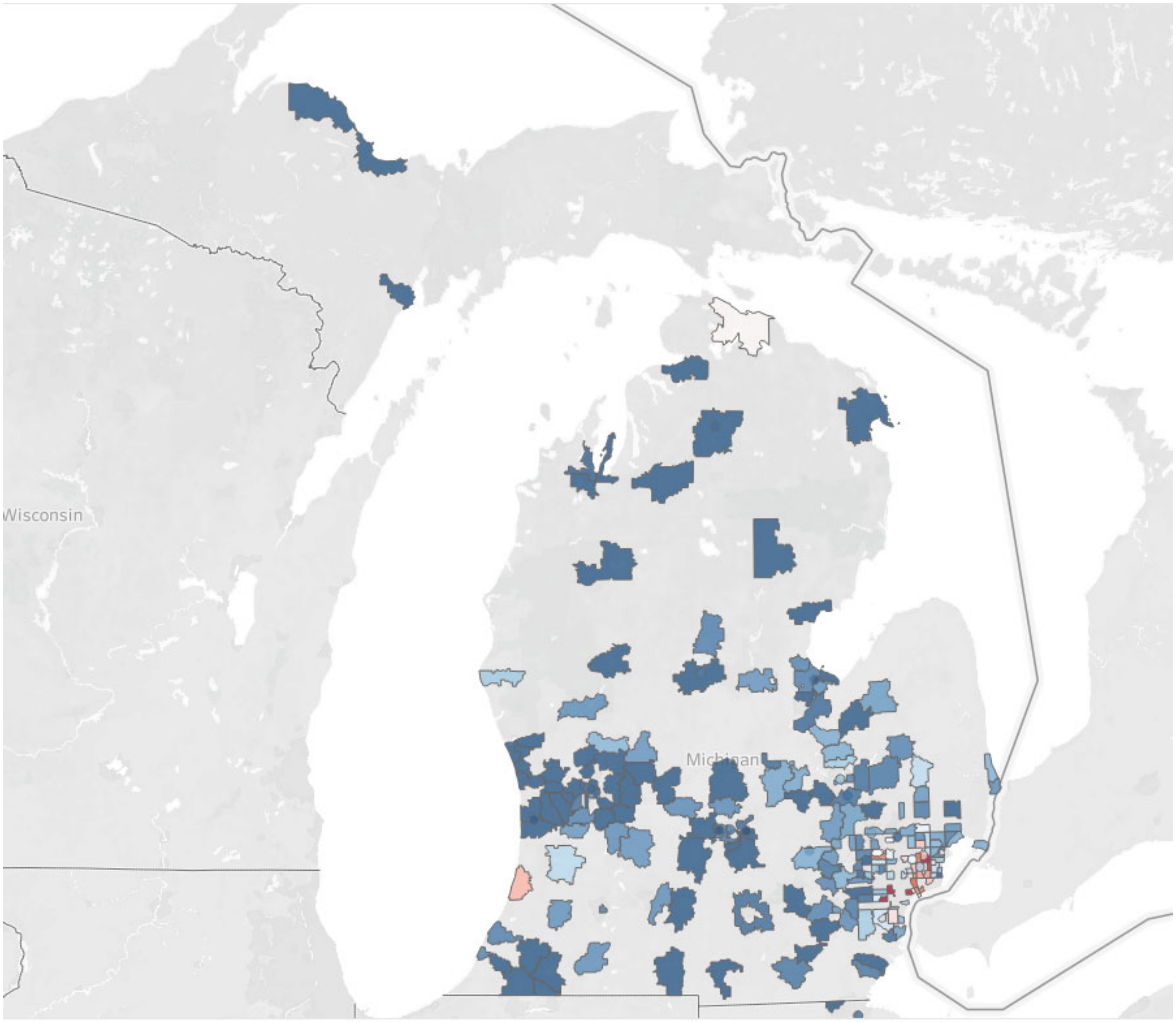


Fig10 – This diagram shows the litigation percent by census tract

The figure above shows the Personal Injury Protection Litigation Percentage based on the Loss Location of the Claims. This graph displays the data on the Zip/Postal Code level. Here we are looking at Zip Codes with at least 10 PIP Claims that have occurred in them and the color represents the percentage of those claims that were litigated. As you can see, the Southeast has a high rate of Claims that end up in litigation, and this is definitely something that the model is picking up. You can see in the figure below that a lot of the problem is also centered in the Sterling Heights/Warren, Michigan area; there seems to be some behavior that is going on in this area that is leading to more of these Claims ending up in litigation. This pattern is picked up by the various predictive models and can be utilized to predict whether or not future Claims will end up in litigation. The Predictive Models are fed Latitude and Longitude features that they correlate and pattern match with the Litigation Target variable.

Geospatial Analysis continued

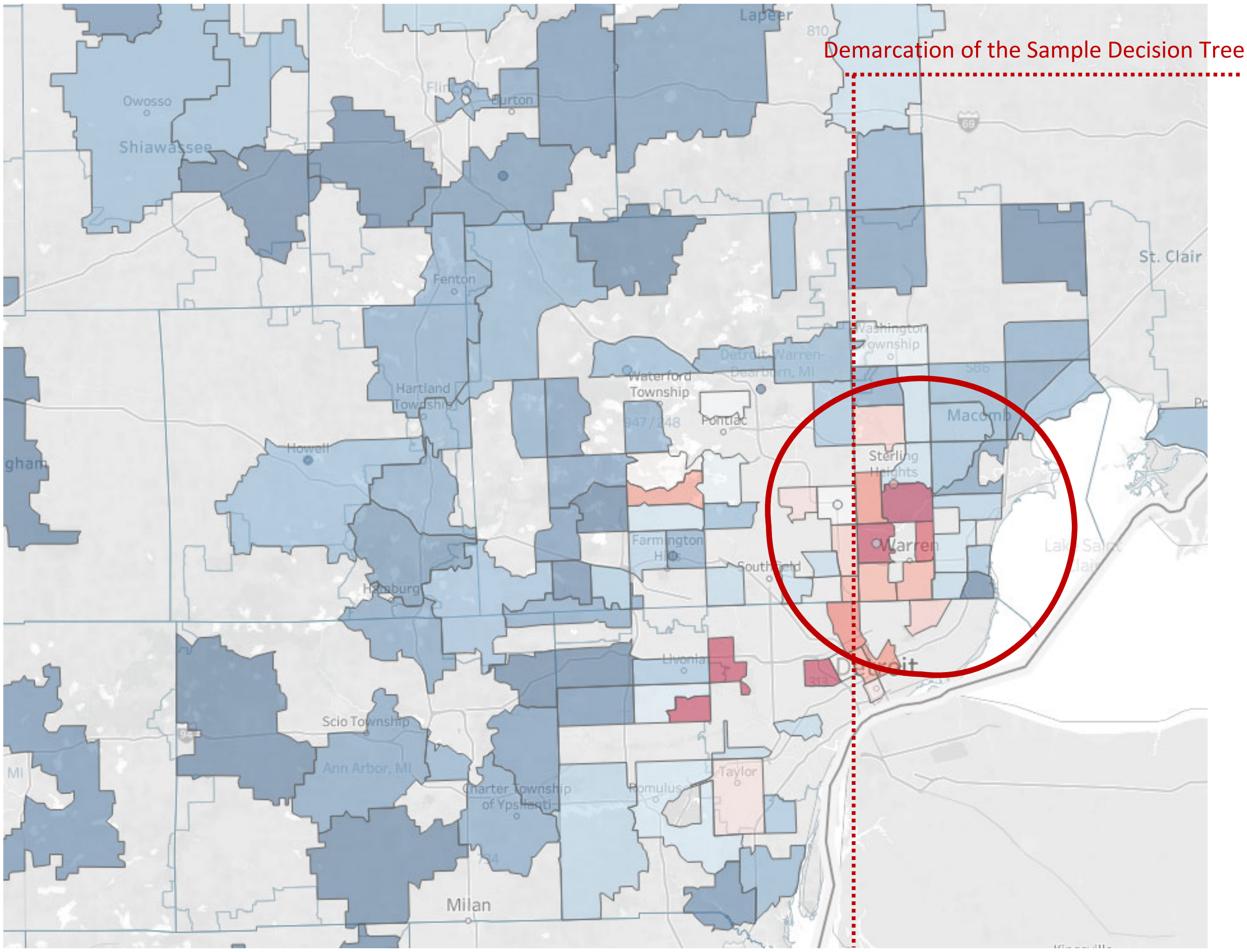
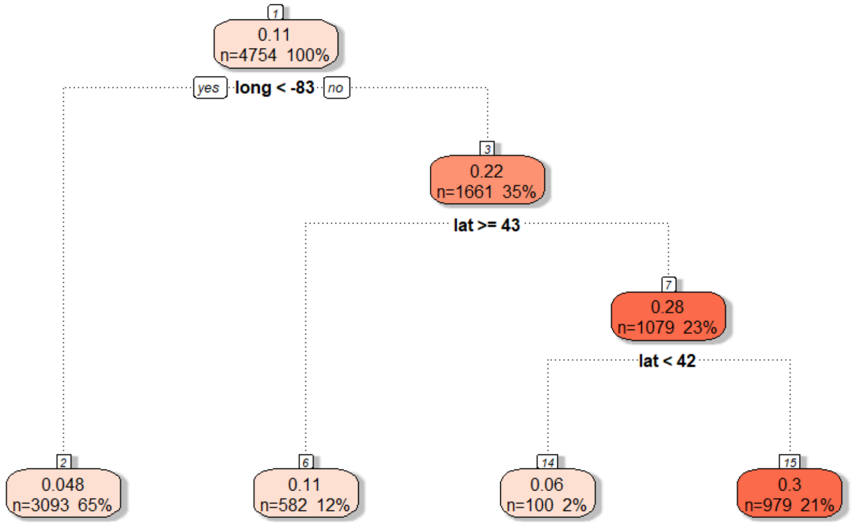

Demarcation of the Sample Decision Tree

Fig 11 - The lines on the graph above so the boundaries that are set up by the decision tree below. The decision tree below is able to differentiate litigation proclivity based on the latitude and longitude data that it was fed. This tree is similar to one that are used to construct the tree based models that are discussed below.



Insured Age as Predictor Variable

I also have included the age of the insured or claimant that is involved in the Claim as a predictor variable, in the instances where there are multiple injured parties in a claim, I took the max of age of any of the injured parties, the reason for this is because if we suspect anyone to injured in an auto claim it would be the oldest of those involved in the accident. We can see from the Graphic below that PIP Claims are more heavily litigated when the insured are ages 60 through 75, this is an interesting pattern in the data, and it may be because this is when drivers begin to slow down mentally and physically and also become more susceptible to injuries.

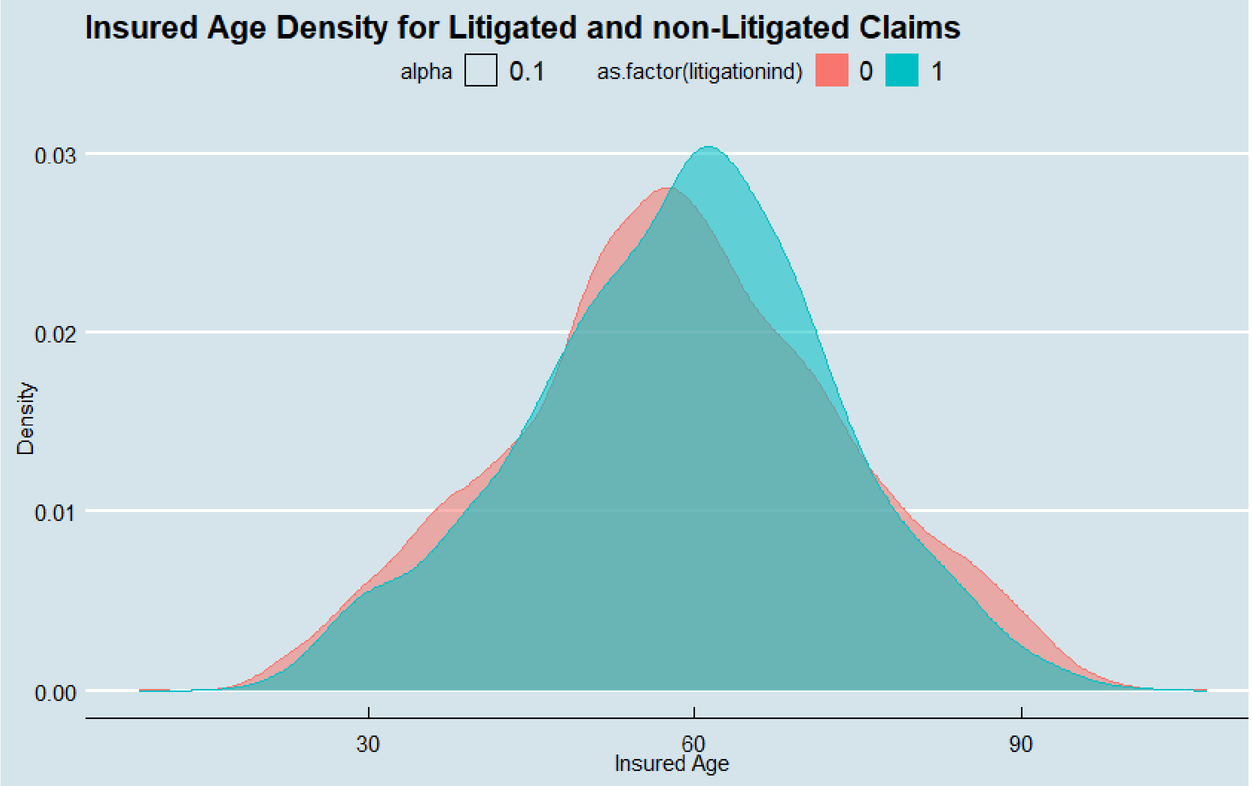


Fig12 – This diagram shows the density of insured age according to litigation

Building the Predictive Model

Once I have done all of the data aggregation gathering and prep, I pull the various data sources into my R Script and aggregate the data together. I have discussed the feature engineering for the predictive model in the sections above.

Imputing Missing Data using MICE

During this initiative, I am bringing in a multitude of data from different sources, and because of this, there is some missing data; not all Claims have all of their relevant data in all of the admin systems. Because of this, I handle these missing values by using an imputation method known as Multiple Imputation Chained Equations[14]. This method performs the imputation multiple times and then determines what the most likely result is based on the aggregation of those multiple imputations.

Constructing the Training and Test Set

When determining what data to use to build the predictive models, often the stratified split is relied upon in machine learning, where the data is split 70% training and 30% testing and all of the available data is randomly sampled while maintaining that the distribution of the target variable remains equal in each of the datasets, meaning in the prevalence of a positive binary target variable is 10% in aggregate then each of the training and test datasets will have 10% of positives for their target variables. However, for many real-life applications, this can fall short because the nature of the data-generating processes changes over time. Meaning that perhaps the data changes dramatically over time and variables shift in their ability to be predictive or the relationships between the variables in the feature space and the target variable. So to be confident that I am building a model that will be generalizable for the future, I choose to have the training data set be PIP claims with Reported Dates in the years 2016, 2017, and 2018. The testing data set is PIP Claims with a reported date between 12-31-2018 and 7-1-2019, the first half of 2019. More recent claims were not used because the nature of the target variable of litigation is that the Claims develop over periods of months or even years, so to determine the true target variable of a claim, I need to give it time to develop.

This methodology for constructing training and testing sets will most likely appear to decrease the performance of my models, but the results will more accurately represent the type of performance I can expect of

Trying out Various Predictive Models

Corey's Random Forest: This is a function that I built myself that allows the user to build a Random Forest using rpart decision trees in R. This model performs similarly to the Ranger Random Forest model and

Logistic Regression Model: The logistic regression model is used to build a baseline that the other models can be compared to. Logistic regression is good for building a model that correlates basic relationships between your feature space and your binary target variable. It is a type of Generalized Linear Model. The shortcomings of the logistic regression model are that it does not take into account interaction relationships between variables in the feature space. It only combines them in a multiplicative manner.

Ranger Random Forest: The Ranger random forest function is used to compare my random forest function to one from another package in R. The Ranger Random Forest Function did outperform my random forest function in this case.

Gradient Boosting Machine (caret): The Gradient Boosting Machine model slightly outperformed both Corey's Random Forest and the ranger random forest functions. The gradient Boosting machine has a boosting or error reduction component to it. It fits a tree-based model based off a subset of the attributes and then measures the error function after this initial model and recursively fits sequential models off of that error function each time fitting new models based off the error of the previous models. It slowly learns how to reduce the error better using all the variables at its disposal. The only real downsides to the gradient boosting Machine are that it can potentially overfit relationships between variables which I was worried about. Still, given the performance of the Gradient Boosting Machine I think that it is not overfitting the feature space with the target variable.

Day One Gradient Boosting Model: In this model, I decided that I would only use data elements that are available in the first day of the Claim; this data includes the loss location, the age of the insured, and the household attributes of the insured. This model performed worse than all the models above but still had solid predictive with a ROC of .79, and this is a model that could be used to triage Claims as soon as they are reported, which would be very helpful.

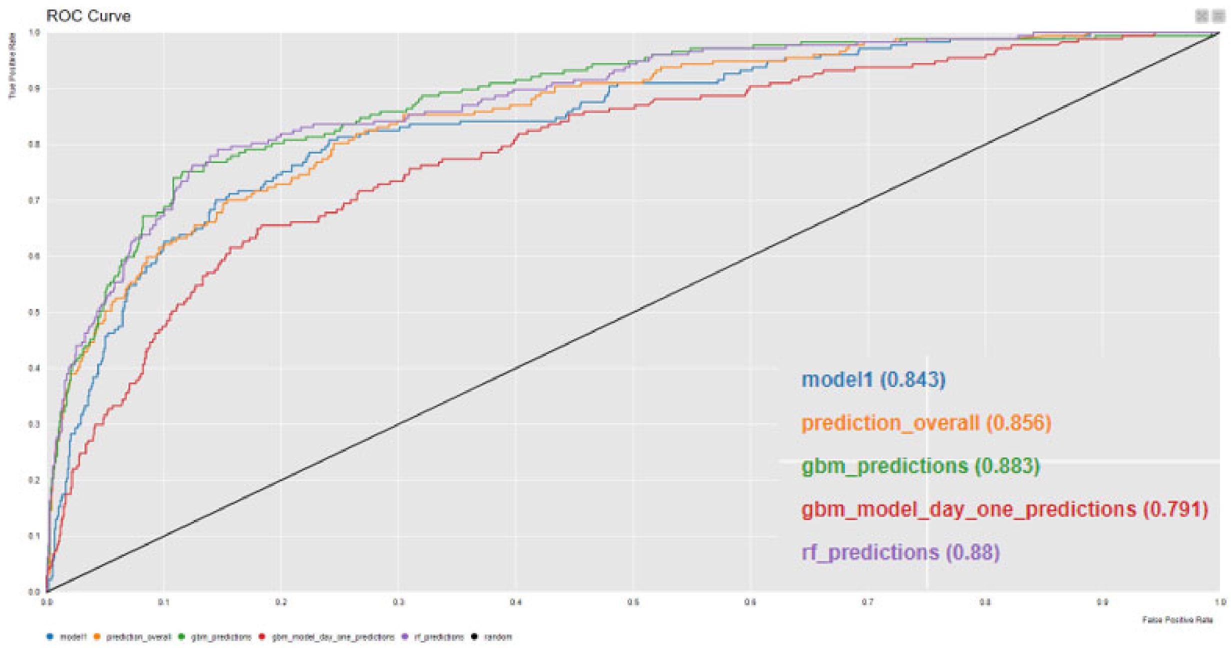


Fig12 – All of the different ROC curves by predictive model

In the graph above, you can see all the various models compared to one another. The lowest performing model is the day-one GBM model, that made predictions on data that is available day one of the Claim, this is expected because this model has much less information to work with. Model1 represents the logistic regression model; this model does comparable to my Random Forest model, which is represented by prediction overall, the two top performing models are the rf_predicitons which is the ranger random forest, and the 60-day Gradient Boosting Machine, both of these models capture patterns in the feature space extremely well. You can see that at the high end of scores both models do extremely well, they capture the first 30% of true positives, while only getting about 2% of false positives.

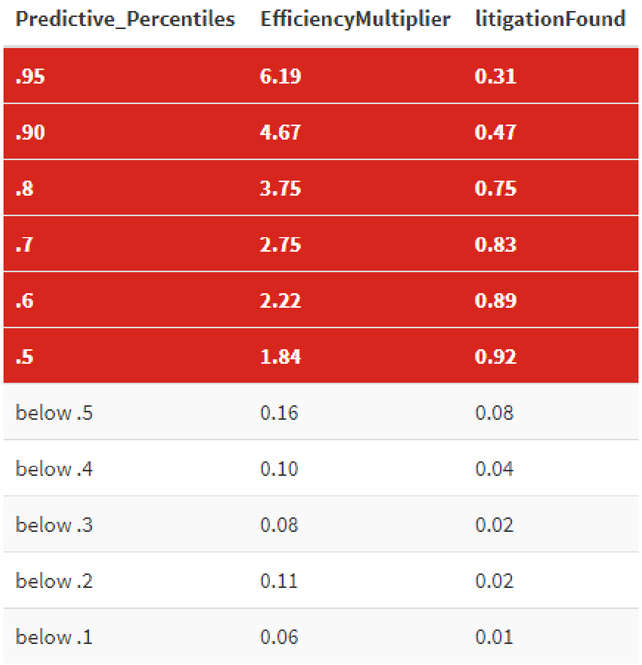
Evaluating the Model Results

Metrics that Make Sense for this Initiative

Even though this is a classification problem, I ran the models as though they were predicting a regression problem; this way, they will return a probability that the Claim will go to litigation. Because it is a relatively unbalanced dataset (11% are positives)

Evaluation

Using just the Process Based Litigation Proclivities metrics and the geographic location of the Claim, the predictive model was able to return a promising result. The chart below shows the results of this model for example the top 5% of highest-scoring claims according to this predictive model represented 32% of the Claims that went to litigation. And the top 50% of highest-scoring claims represented 94% of the Claims that went to litigation.



Predictive_Percentiles	EfficiencyMultiplier	litigationFound
.95	**6.19**	**0.31**
.90	**4.67**	**0.47**
.8	**3.75**	**0.75**
.7	**2.75**	**0.83**
.6	**2.22**	**0.89**
.5	**1.84**	**0.92**
below .5	0.16	0.08
below .4	0.10	0.04
below .3	0.08	0.02
below .2	0.11	0.02
below .1	0.06	0.01

Fig13 – Predictive model performance

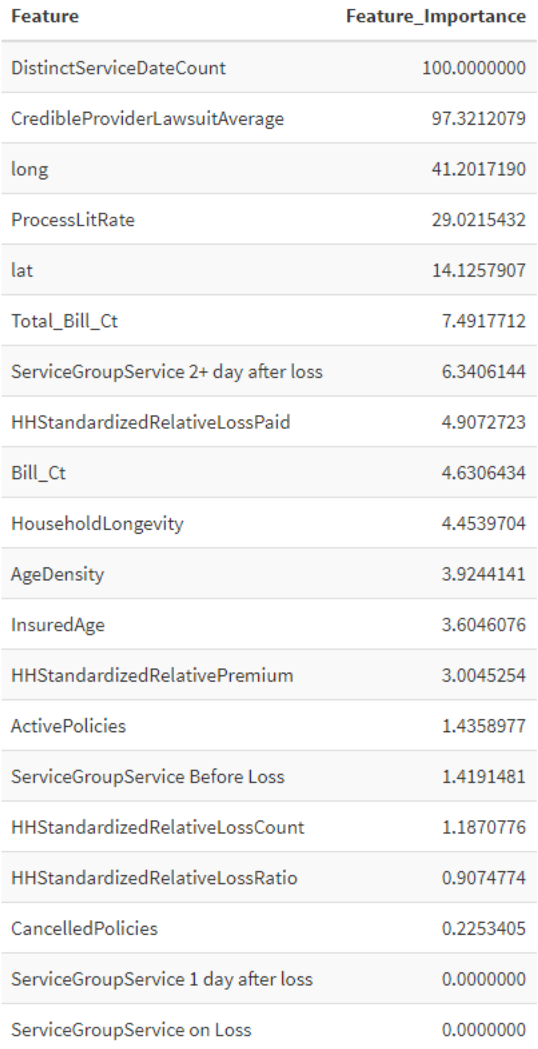
The table on the left gives us an idea of the different model performances at the different score intervals the higher scoring a claim is the more likely that the claim will end up in litigation according to the model. We can see that the top 5% of highest scoring claims account for 31% of the litigation in the test set. This means that trying to find litigation by looking at claims in this range is 6.19 times more efficient that random sampling of the entire test set.

The benefit of being able to identify Claims that are likely to go to litigation before they actually do is that those Claims can be reassigned to Adjusters that are more equipped to deal with tricky Claims and potentially prevent this Claims from ever going to litigation or at least limit the amount that we pay out on this Clams. This model allows the insurance company to put the right Claims in the hands of experts.

Once implementation time comes I think it may make sense to have multiple models running at once, there may be a model that scores Claims right when they are reported and another model that scores them after the first month, two months, etc.… These details still need to be ironed out.

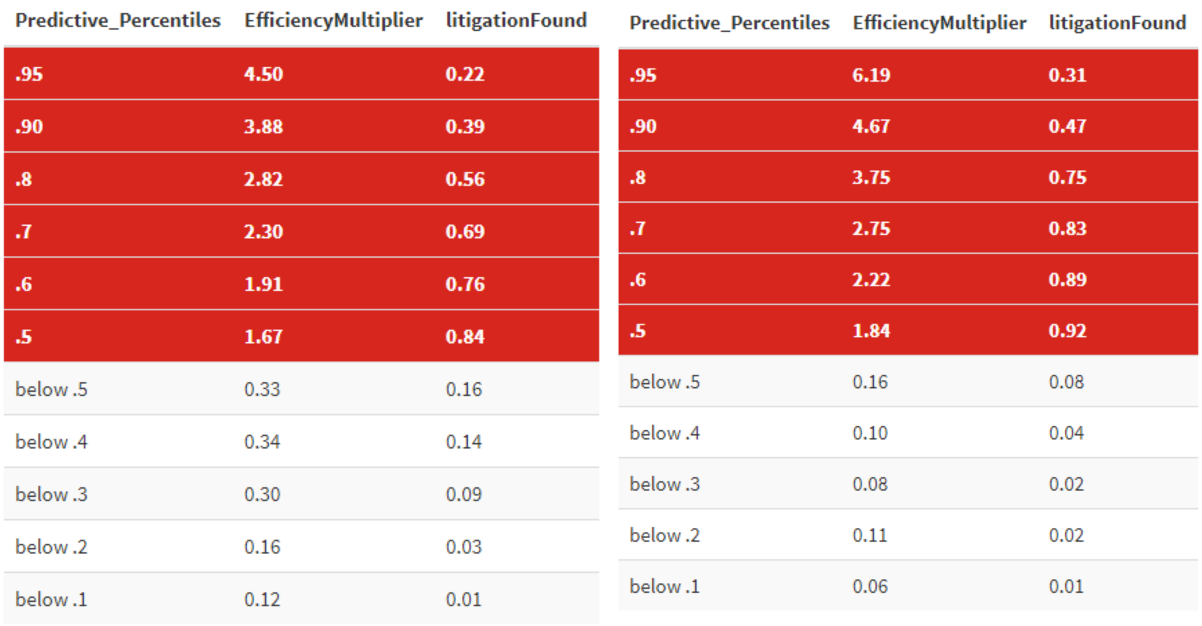
Variable Importance

The table to the right displays the variable importance for the predictive model, the top variable distinct service date counts represents the number of days that the insured got service in the first 60 days following the report of the Claim. This seems to be very predictive and is highly correlated to the target variable, meaning that the more medical service dates that someone has the more likely that claim is to become litigated. The credible provider lawsuit average is one of the variables I describe in the feature engineering section above, and attempts to predict whether or not a claim will go to litigation based on the previous claims we have had with that medical provider, it is promising to see this variable perform so well. We can see that latitude and longitude were also very influential variables, the location of the loss is very important when predicting PIP Litigation, this is a variable that is also available from the start of the claim, which means it can be used for a model at any point in the Claim lifecycle.



Feature	Feature_Importance
DistinctServiceDateCount	100.0000000
CredibleProviderLawsuitAverage	97.3212079
long	41.2017190
ProcessLitRate	29.0215432
lat	14.1257907
Total_Bill_Ct	7.4917712
ServiceGroupService 2+ day after loss	6.3406144
HHStandardizedRelativeLossPaid	4.9072723
Bill_Ct	4.6306434
HouseholdLongevity	4.4539704
AgeDensity	3.9244141
InsuredAge	3.6046076
HHStandardizedRelativePremium	3.0045254
ActivePolicies	1.4358977
ServiceGroupService Before Loss	1.4191481
HHStandardizedRelativeLossCount	1.1870776
HHStandardizedRelativeLossRatio	0.9074774
CancelledPolicies	0.2253405
ServiceGroupService 1 day after loss	0.0000000
ServiceGroupService on Loss	0.0000000

Fig14 – Variable Importance



Day One Model vs Day 60 Model

Predictive_Percentiles	EfficiencyMultiplier	litigationFound
.95	4.50	0.22
.90	3.88	0.39
.8	2.82	0.56
.7	2.30	0.69
.6	1.91	0.76
.5	1.67	0.84
below .5	0.33	0.16
below .4	0.34	0.14
below .3	0.30	0.09
below .2	0.16	0.03
below .1	0.12	0.01

Predictive_Percentiles	EfficiencyMultiplier	litigationFound
.95	6.19	0.31
.90	4.67	0.47
.8	3.75	0.75
.7	2.75	0.83
.6	2.22	0.89
.5	1.84	0.92
below .5	0.16	0.08
below .4	0.10	0.04
below .3	0.08	0.02
below .2	0.11	0.02
below .1	0.06	0.01

Fig15 – Comparing day one and day 60 models

On the left in the figure above you can see the performance on left reflects the model built using only data available on day one of the Claim. You can see that it performs many times better than randomly selecting. This model only depends on loss location and previous information that we know about the insured such as their loss history and their age. The Day one model has significantly worse performance but could be implemented right away.

Summary:

The initiative to predict which Personal Injury Protection Claims will end up in litigation was very successful; the metrics above show that the predictive model performs many multiples better than random. This predictive model will be utilized in the Claims Process to help reassign Claims to appropriate resources. This will help put Claims in the hands of people that are best suited to handle them. Other Predictive Models or business rules based on Predictive Models may also be utilized earlier in the Claim Process. Farm Bureau's Claims department could also utilize a model that helps triage Claims from the onset of the Claim, this could be done with existing data elements or we could pursue new technological improvements that would help facilitate triage earlier in the Claim such as the methods described in [15]. The best approach seems to be multi-faceted, with different predictive models at points in the process when they can be best utilized. Once Claims that are likely to be Litigated are identified then, the optimal path for these Claims can be determined. This may include certain things like choosing what bills to pay, what attorneys to hire, or what adjusters should own these potentially harmful claims. There are also other actions that can be taken on certain Claims things such as Independent Medical Examinations that will inform us whether an insured's injuries are legitimate or if they are being exaggerated. These different Claim Handling actions are taken today, but they are more on a one-off basis and not a fully coordinated effort. The predictive models will help identify the correct way to handle the claims, but these actions will still need to be executed by the various members of the Farm Bureau Claims Department.

Process Conformance

For the effect of the Predictive Models to be realized, there needs to be some level of process control to make sure that the strategy of handling Claims is executed. To do this, we can also start to leverage Process Mining and Business Intelligence.

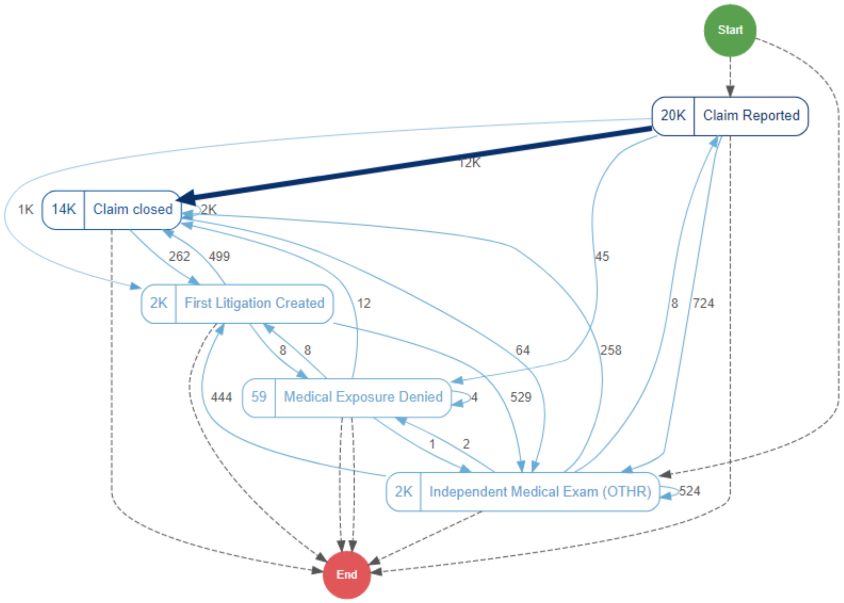


Fig16 – Process Mining Dashboard

Fig9 This is a picture of the PIP Process Dashboard that I am developing that is part of our tableau environment here at Farm Bureau. It is able to show Claim Paths and any number of Claim related metrics. Once fully fledged Claim handling strategies are developed, dashboards such as this will be able to utilized to ensure that the processes are being executed in the correct way. Any number of activities can be included on these process mining dashboards and they offer the ability to manage any aspect of the Business Process.

Tools such as the Process Dashboard described above will help Farm Bureau execute on their Claim Handling Strategies.

References

[1] W. M. P. van der Aalst, Process Mining - Discovery, Conformance and Enhancement of Business Processes, 1st Edition, Springer, 2011.

[2] Polato, M., Sperduti, A., Burattin, A., de Leoni, M.: **Time and activity sequence prediction of business process instances**. arXiv preprint arXiv:1602.07566 (2016)

https://arxiv.org/pdf/1602.07566.pdf

[3] G. T. Lakshmanan, D. Shamsi, Y. N. Doganata, M. Unuvar, R. Khalaf, **A markov prediction model for data-driven semi-structured business processes**, Knowledge and Information Systemsdoi:10.1007/ s10115-013-0697-8.

[4] Widad Es Soufi, Esma Yahia, Lionel Roucoules. On the use of Process Mining and Machine Learning to support decision making in systems design. 13th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2016, Columbia, United States. pp.56-66, ff10.1007/978-3-319-54660-5_6ff. ffhal-01403073fhttps://hal.archives-ouvertes.fr/hal-01403073/document

[5] Process Mining in Insurance: Measuring and Managing Activity Costs
https://www.casact.org/community/affiliates/maf/0919/1.pdf

[6] M. de Leoni, et al., A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs, Information Systems (2015),
http://dx.doi.org/10.1016/j.is.2015.07.003i

[7] A Network-Based Approach to Modeling and Predicting Product Coconsideration Relations

[8] Carlos Andre Reis Pinheiro, Oi, Rio de Janeiro, Brazil. Highlighting Unusual Behavior in Insurance Based on Social Network Analysis
http://support.sas.com/resources/papers/proceedings11/130-2011.pdf

[9] M. Pospíšil, V. Mates, and T. Hruška, "Process Mining in Manufacturing Company," in The Fifth International Conference on Information, Process, and Knowledge Management, Nice, France, IARIA, 2013, pp. 143-148, ISBN 978-1-61208-254-7

[10] Predicting Insurance Fraud with Machine Learning (SMOTE)

https://medium.com/analytics-vidhya/predicting-insurance-fraud-with-machine-learning-smote-da94adf8fb62

[11] Artificial Intelligence and Process Mining

https://medium.com/datadriveninvestor/artificial-intelligence-in-process-mining-d8a61c0adfd1

[12] "Process mining on the loan application process of a Dutch Financial Institute. BPI Challenge 2017" Liese Blevi, Lucie Delporte, Julie Robbrecht KPMG Technology Advisory, Bourgetlaan 40, 1130 Brussels, Belgium

[13] *"Predictive Business Process Monitoring with LSTM Neural Networks”* **by Niek Tax, Ilya Verenich, Marcello La Rosa and Marlon Dumas**

[14] Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). **Multiple imputation by chained equations: what is it and how does it work?.** *International journal of methods in psychiatric research*, *20*(1), 40–49. https://doi.org/10.1002/mpr.329

[15] SD Kidd, DA Smith - US Patent 7,698,086, 2010 - Google Patents. **Method and apparatus for obtaining and using event data recorder triage data**

Appendices

https://github.com/crarnouts/Data-698-Thesis/blob/main/Github-Predictive-Model-Version.R

https://raw.githubusercontent.com/crarnouts/Data-698-Thesis/main/Github-Predictive-Model-Version.R

```
require(ISLR)
library(tidyverse)
require(gridExtra)
library(Amelia)
library(kableExtra)
library(caret)
library(DMwR)
library(scales)
library(purrr)
library(RColorBrewer)
library(ROCR)
library(corrplot)
library(digest)
library(openssl)

train <- read.csv("C:/Users/carnout/Documents/train_data.csv")
test <- read.csv("https://raw.githubusercontent.com/crarnouts/Data-698-Thesis/main/test_data.csv")
household_attributes <- read.csv("https://raw.githubusercontent.com/crarnouts/Data-698-Thesis/main/household_attributes_data2.csv")
insured_age <- read.csv("https://raw.githubusercontent.com/crarnouts/Data-698-Thesis/main/insured_age_data.csv")
losslocation <- read.csv("https://raw.githubusercontent.com/crarnouts/Data-698-Thesis/main/losslocation_data.csv")
billing_attributes <- read.csv("https://raw.githubusercontent.com/crarnouts/Data-698-Thesis/main/billing_attributes_data.csv")
```

```
ProviderMeanEncodingTest <- read.csv("https://raw.githubusercontent.com/crarnouts/Data-698-Thesis/main/ProviderMeanEncodingTest_data.csv")

ProviderMeanEncodingTrain <- read.csv("https://raw.githubusercontent.com/crarnouts/Data-698-Thesis/main/ProviderMeanEncodingTrain_data.csv")

billing_attributes <- billing_attributes %>%
select(ClaimNumber,Bill_Ct,Total_Bill_Ct,DistinctServiceDateCount,ServiceGroup)

household_attributes <- household_attributes %>%
select(ClaimNumber,HHStandardizedRelativePremium,HHStandardizedRelativeLossRatio

,HHStandardizedRelativeLossCount,HHStandardizedRelativeLossPaid,HouseholdLongevity

                                        ,ActivePolicies,CancelledPolicies,YearlyTotalPremium)

train <- merge(train,losslocation)

test <- merge(test,losslocation)

 train <- merge(train,ProviderMeanEncodingTrain, all.x = TRUE)

 test <- merge(test,ProviderMeanEncodingTest, all.x = TRUE)


train <- merge(train,billing_attributes, all.x = TRUE)

test <- merge(test,billing_attributes, all.x = TRUE)


train <- merge(train,household_attributes, all.x = TRUE)

test <- merge(test,household_attributes, all.x = TRUE)


train <- merge(train,insured_age, all.x = TRUE)

test <- merge(test,insured_age, all.x = TRUE)


### Encoding for Claims that don’t have billing attributes


train$Bill_Ct[is.na(train$Bill_Ct)] <- 0

train$Total_Bill_Ct[is.na(train$Total_Bill_Ct)]<-0

train$DistinctServiceDateCount[is.na(train$DistinctServiceDateCount)]<-0
```

```
train$ServiceGroup[is.na(train$ServiceGroup)]<- "No Medical Review in Firsts 60 Days"

test$Bill_Ct[is.na(test$Bill_Ct)] <- 0

test$Total_Bill_Ct[is.na(test$Total_Bill_Ct)]<-0

test$DistinctServiceDateCount[is.na(test$DistinctServiceDateCount)]<-0

test$ServiceGroup[is.na(test$ServiceGroup)]<- "No Medical Review in Firsts 60 Days"



# Using the MICE package to impute for any NULL Values

 library(mice)

 imputed_Data <- mice(train,method = 'cart')

 imputed_Data2 <- mice(test,method = 'cart')

 train <- complete(imputed_Data,2)

 test <- complete(imputed_Data2,2)

 # mean imputation

 for(i in 1:ncol(train)) {

   train[ , i][is.na(train[ , i])] <- mean(train[ , i], na.rm = TRUE)

 }


 # mean imputation

 for(i in 1:ncol(test)) {

   test[ , i][is.na(test[ , i])] <- mean(test[ , i], na.rm = TRUE)

 }




### Lets try out the ol random forest ##

#################### BRING IN THE RANDOM FOREST FUNCTION #############################
```

```
source("https://raw.githubusercontent.com/crarnouts/Random_Forest_Function/master/RandomForestNulls_testing.R")

source("https://raw.githubusercontent.com/crarnouts/Corey-s_Scripts_For_Reference/master/Density_diff_2.R")

###########################################################################################


 # Renaming Process Variables so that are more intuitive

 train <- train %>%

  dplyr::rename(

    ProcessLitRate = AVGRate,

    CredProcessLitRate = AVGCredRate

  )


test <- test %>%

  dplyr::rename(

    ProcessLitRate = AVGRate,

    CredProcessLitRate = AVGCredRate

  )




## create training dataset that will be used fot the 60 day model

train2 <- train %>%
dplyr::select(lat,long,ProcessLitRate,litigationind,CredibleProviderLawsuitAverage,Bill_Ct

                              ,Total_Bill_Ct,DistinctServiceDateCount,ServiceGroup

                              ,HHStandardizedRelativePremium,HHStandardizedRelativeLossRatio


,HHStandardizedRelativeLossCount,HHStandardizedRelativeLossPaid,HouseholdLongevity

                              ,ActivePolicies,CancelledPolicies,ActivePolicies,CancelledPolicies,InsuredAge)
```

```
## create training dataset that will be used for the day 1 model

train3 <- train %>% dplyr::select(lat,long,litigationind

                              ,HHStandardizedRelativePremium,HHStandardizedRelativeLossRatio


,HHStandardizedRelativeLossCount,HHStandardizedRelativeLossPaid,HouseholdLongevity

                              ,ActivePolicies,CancelledPolicies,ActivePolicies,CancelledPolicies,InsuredAge)
```

```
## A look at the insured age variable #####

library(ggplot)

library(ggthemes)

ggplot(train2,aes(InsuredAge))+geom_density(aes(color=as.factor(litigationind),fill=as.factor(litigationind),alpha=0.1))+

  labs(x="Insured Age",y="Density",title="Insured Age Density for Litigated and non-Litigated Claims")+

  theme(legend.position = "top")+theme_economist()
```

```
## Create a Logistic Regression model to establish a baseline model


model1 <- glm(litigationind ~ ProcessLitRate+CredibleProviderLawsuitAverage

             +DistinctServiceDateCount+lat+long+Total_Bill_Ct+ServiceGroup,

             family = binomial(link = "logit"),

             train)

summary(model1)


test$model1 <- predict.glm(model1, test,"response")
```

```
cor(test$model1,test$litigationind)


## Day One logistic regression model


model2 <- glm(litigationind ~ .,
            family = binomial(link = "logit"),
            train3)
summary(model2)


test$model2 <- predict.glm(model2, test,"response")


cor(test$model2,test$litigationind)




## Make predictions on the test dataset using the Random forest function that I created
test <- RF_with_Nulls(train2,test,"litigationind",.5,4,100,.003,10,300)



## Create a Factor version of the target variable


test$litigationindfact <- as.factor(test$litigationind)


################### Gradient Boosting Model ########################################


# create the gradient boosting Machine with caret package and utilize the cross validation
library(gbm)
```

```
set.seed(42)

gbm_model <- train(litigationind  ~ ., data = train2, method="gbm", verbose = FALSE,

                trControl = trainControl("cv", number = 5))




gbm_model


# Make predictions using the gbm_model on the test dataset

test$gbm_predictions <- predict(gbm_model, test)


# Day one model with GBM


set.seed(42)

gbm_model_day_one <- train(litigationind  ~ ., data = train3, method="gbm", verbose = FALSE,

                trControl = trainControl("cv", number = 5))




gbm_model_day_one


# Make predictions using the gbm_model on the test dataset

test$gbm_model_day_one_predictions <- predict(gbm_model_day_one, test)


######################################################################################
##########

########### Random Forest Function from Ranger
#################################################
```

```
rf_model <- train(litigationind  ~ ., data = train2, method = "ranger",

                scale = TRUE,

                trControl = trainControl("cv", number = 3))

test$rf_predictions <- predict(rf_model, test)





######################################################################################


library(pROC)

roc(test$litigationindfact, test$model1, plot = TRUE, legacy.axes = TRUE, print.auc = TRUE)



roc(test$litigationindfact, test$prediction_overall, plot = TRUE, legacy.axes = TRUE, print.auc = TRUE)



roc(test$litigationindfact, test$gbm_predictions, plot = TRUE, legacy.axes = TRUE, print.auc = TRUE)


roc(test$litigationindfact, test$gbm_model_day_one_predictions, plot = TRUE, legacy.axes = TRUE,
print.auc = TRUE)


roc(test$litigationindfact, test$rf_predictions, plot = TRUE, legacy.axes = TRUE, print.auc = TRUE)
```

```
# roc(test$litigationindfact, test$model2, plot = TRUE, legacy.axes = TRUE, print.auc = TRUE)

################################## Assemble a table with the best model #################



nintyfifth_percentile <- mean(test$litigationind[test$gbm_predictions> quantile(test$gbm_predictions,
.95)])/mean(test$litigationind)

nintypercentile <-mean(test$litigationind[test$gbm_predictions> quantile(test$gbm_predictions,
.90)])/mean(test$litigationind)

eightypercentile <-mean(test$litigationind[test$gbm_predictions> quantile(test$gbm_predictions,
.80)])/mean(test$litigationind)

seventypercentile <-mean(test$litigationind[test$gbm_predictions> quantile(test$gbm_predictions,
.70)])/mean(test$litigationind)

sixtypercentile <-mean(test$litigationind[test$gbm_predictions> quantile(test$gbm_predictions,
.60)])/mean(test$litigationind)
fiftypercentile <-mean(test$litigationind[test$gbm_predictions> quantile(test$gbm_predictions,
.50)])/mean(test$litigationind)

belowfiftypercentile <-mean(test$litigationind[test$gbm_predictions< quantile(test$gbm_predictions,
.50)])/mean(test$litigationind)

belowfortypercentile <-mean(test$litigationind[test$gbm_predictions< quantile(test$gbm_predictions,
.40)])/mean(test$litigationind)

belowthirtypercentile <-mean(test$litigationind[test$gbm_predictions< quantile(test$gbm_predictions,
.30)])/mean(test$litigationind)
```

```
belowtwentypercentile <-mean(test$litigationind[test$gbm_predictions<
quantile(test$gbm_predictions, .20)])/mean(test$litigationind)


belowtenthpercentile <-mean(test$litigationind[test$gbm_predictions< quantile(test$gbm_predictions,
.10)])/mean(test$litigationind)




litigationFound95 <- .05* nintyfifth_percentile

litigationFound90<- .1 * nintypercentile

litigationFound80 <- .2*eightypercentile

litigationFound70 <- .3* seventypercentile

litigationFound60 <- .4*sixtypercentile

litigationFound50<- .5*fiftypercentile

litigationFoundbelow50<- .5*belowfiftypercentile

litigationFoundbelow40<- .4*belowfortypercentile

litigationFoundbelow30<- .3*belowthirtypercentile

litigationFoundbelow20<- .2*belowtwentypercentile

litigationFoundbelow10<- .1*belowtenthpercentile


library(data.table)


dt <- data.table(x = c(".95", ".90",".8",".7",".6",".5","below .5","below .4","below .3","below .2","below
.1"), y =
c(nintyfifth_percentile,nintypercentile,eightypercentile,seventypercentile,sixtypercentile,fiftypercentile,
belowfiftypercentile,belowfortypercentile,belowthirtypercentile,belowtwentypercentile,belowtenthper
centile),

           z
=c(litigationFound95,litigationFound90,litigationFound80,litigationFound70,litigationFound60,


litigationFound50,litigationFoundbelow50,litigationFoundbelow40,litigationFoundbelow30,litigationFou
ndbelow20,litigationFoundbelow10))
```

```
dt$Predictive_Percentiles <- dt$x

dt$EfficiencyMultiplier <- dt$y

dt$litigationFound <- dt$z

dt$x <- NULL

dt$y <- NULL

dt$z <- NULL


dt$EfficiencyMultiplier <- format(round(dt$EfficiencyMultiplier, 2), nsmall = 2)

dt$litigationFound <- format(round(dt$litigationFound, 2), nsmall = 2)



kable(dt) %>%

  kable_styling("striped", full_width = F) %>%

  row_spec(1:6, bold = T, color = "white", background = "#D7261E")


#################### DAY ONE MODEL
##############################################################



nintyfifth_percentile <- mean(test$litigationind[test$gbm_model_day_one_predictions>
quantile(test$gbm_model_day_one_predictions, .95)])/mean(test$litigationind)


nintypercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions>
quantile(test$gbm_model_day_one_predictions, .90)])/mean(test$litigationind)


eightypercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions>
quantile(test$gbm_model_day_one_predictions, .80)])/mean(test$litigationind)
```

```
seventypercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions>
quantile(test$gbm_model_day_one_predictions, .70)])/mean(test$litigationind)


sixtypercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions>
quantile(test$gbm_model_day_one_predictions, .60)])/mean(test$litigationind)

fiftypercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions>
quantile(test$gbm_model_day_one_predictions, .50)])/mean(test$litigationind)


belowfiftypercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions<
quantile(test$gbm_model_day_one_predictions, .50)])/mean(test$litigationind)


belowfortypercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions<
quantile(test$gbm_model_day_one_predictions, .40)])/mean(test$litigationind)


belowthirtypercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions<
quantile(test$gbm_model_day_one_predictions, .30)])/mean(test$litigationind)


belowtwentypercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions<
quantile(test$gbm_model_day_one_predictions, .20)])/mean(test$litigationind)


belowtenthpercentile <-mean(test$litigationind[test$gbm_model_day_one_predictions<
quantile(test$gbm_model_day_one_predictions, .10)])/mean(test$litigationind)




litigationFound95 <- .05* nintyfifth_percentile

litigationFound90<- .1 * nintypercentile

litigationFound80 <- .2*eightypercentile

litigationFound70 <- .3* seventypercentile

litigationFound60 <- .4*sixtypercentile

litigationFound50<- .5*fiftypercentile

litigationFoundbelow50<- .5*belowfiftypercentile
```

```
litigationFoundbelow40<- .4*belowfortypercentile

litigationFoundbelow30<- .3*belowthirtypercentile

litigationFoundbelow20<- .2*belowtwentypercentile

litigationFoundbelow10<- .1*belowtenthpercentile


library(data.table)


dt <- data.table(x = c(".95", ".90",".8",".7",".6",".5","below .5","below .4","below .3","below .2","below
.1"), y =
c(nintyfifth_percentile,nintypercentile,eightypercentile,seventypercentile,sixtypercentile,fiftypercentile,
belowfiftypercentile,belowfortypercentile,belowthirtypercentile,belowtwentypercentile,belowtenthper
centile),

                z
=c(litigationFound95,litigationFound90,litigationFound80,litigationFound70,litigationFound60,


litigationFound50,litigationFoundbelow50,litigationFoundbelow40,litigationFoundbelow30,litigationFou
ndbelow20,litigationFoundbelow10))


dt$Predictive_Percentiles <- dt$x

dt$EfficiencyMultiplier <- dt$y

dt$litigationFound <- dt$z

dt$x <- NULL

dt$y <- NULL

dt$z <- NULL


dt$EfficiencyMultiplier <- format(round(dt$EfficiencyMultiplier, 2), nsmall = 2)

dt$litigationFound <- format(round(dt$litigationFound, 2), nsmall = 2)



kable(dt) %>%

  kable_styling("striped", full_width = F) %>%
  row_spec(1:6, bold = T, color = "white", background = "#D7261E")



#####################################################################################
################

############## Variable Importance

varimportance <- varImp(gbm_model)$importance %>%

  as.data.frame() %>%

  rownames_to_column() %>%

  arrange(Overall)


colnames(varimportance) <- c("Feature","Feature_Importance")


varimportance <- varimportance[order(-varimportance$Feature_Importance),]


kable(varimportance)%>%

  kable_styling("striped", full_width = F)
```