British Columbia Judicial Decisions Analysis

UBC Master of Data Science - Computational Linguistics

By: Ilana Zimmerman, Niki Hajmoshir, Ravi Gill

Capstone project supervisor: Julian Brooke

Client: Peter A. Allard School of Law , Lachlan Caunt

Acknowledgments

To our supervisor—Dr. Julian Brooke—we would have been not be able to do this without you.

Thank you for your patient guidance, enthusiastic encouragement and useful critiques of this research work.

To our capstone client — Lachlan Caunt — who brought this amazing project to us and did not hesitate to help and give directions.

Your willingness to give your time so generously has been very much appreciated.

British Columbia Judicial Decisions Analysis Directory

Summary

In recent years, some members of the legal community specializing in negligence cases have begun to suspect that the overall damage payouts have been steadily increasing in British Columbia. Negligence can be described as “an area of tort law that deals with the breach of duty to take care and it involves harm caused by carelessness, not intentional harm.”[1]. Negligence cases may involve damage payments to the plaintiff broken down into specific sub-categories based on the evidence presented and the precedence set for similar cases that have occurred in the past. They also may find the plaintiff partially liable for the damages which is known as contributory negligence. The goal of this project is to analyze a sample of negligence cases in BC since the year 2000 in order to determine how damages and contributory negligence have been changing over time. Currently, there does not exist any research into whether damages or contributory negligence decisions have been changing over time or if they have remained relatively steady.

In order to complete a meaningful analysis, important pieces of information such as the damages paid and the percentage that the plaintiff is found liable must be extracted from case reports, this is the most challenging aspect of our analysis and is referred to as information extraction. Our team focussed on two specific methods to extract relevant information from case reports. The first method is commonly referred to as a rule based approach in natural language processing circles. This method involves laying out specific patterns that a computer will use to search through large amounts of text data and return to the user only the text which matched a pattern. Many case reports will use similar language when stating the final amount of liability or damages awarded which makes the rule based approach a good starting point. The second approach used can be known as either a feature based approach or a classification based approach. The idea behind a classification based approach is to get a computer to inspect every potentially relevant piece of information and determine the group or class the information belongs to. In the context of a negligence case, the class could be the different types of damages that can be awarded such as special, punitive, or aggravated damages. This approach requires some cases to be manually inspected and classified by humans in order for the system to learn to distinguish classes. This is a time consuming process but will typically produce results at least as good as a rule based approach.

The next step is to statistically analyze the results of the information we were able to extract from the case reports. It was found that, on average, negligence cases in BC have been seeing a gradual increase in damage payouts, adjusted for inflation, since the year 2000. It was also found that judges have generally become more thorough in case reports as the average decision length of cases has also been steadily increasing over the same timeframe. Contributory negligence showed a mean of ~ 38% reduction with a steady trend over the years. The analysis has methodically confirmed the overall suspicion in the legal community about the changes in how judges have been treating negligence cases over the past 20 years. It can be inferred by data from the past 20 years that the overall upward trend will continue to rise in the near future.

[1] Negligence Definition

Background Knowledge

In order to fully understand the language, concepts and methodology used in this paper some terminology and concepts need to be introduced.

Damage Types

Tort law is divided into three categories of “Intentional tort”, “Negligence”, and “Strict liability”. In this paper we will only analyse negligence cases in BC since the year 2000. The damages awarded in negligence cases branch into 3 main categories: “non pecuniary”, “pecuniary”, and “other”. Non pecuniary damages are awarded to compensate the plaintiff for things that have no bill or real cost but have a real impact on the plaintiff's well being, an example of this is pain and suffering. Pecuniary damages are damages awarded to the plaintiff where a bill exists or there is a reasonable estimable cost for them. Pecuniary damages can be further separated into:

- General Damages: Any compensation decided by court for any future costs such as future loss of earning
- Special damages: Damages where there is a physical bill has already been produced or previous costs can be calculated. Examples include physiotherapy bills or wage loss due to missed work.

The other category of damages involves:

- Aggravated Damages: Damages to compensate the plaintiff for loss of dignity, and
- Punitive Damages: Damages to punish the defendant for conduct that was malicious, high handed or particularly offensive.

A full flow diagram of damage types are attached in appendix 1.

Technical Knowledge

In this paper two main approaches were used to extract desired outputs as stated in the summary. In the rule based method one tries to assign a set of rules to find patterns and characters that are repeated in the text which may be a marker of the desired information. For example, the text may mention *“I award the plaintiff $20,000 in special damages”,* the rule can be if the word “special damages” occurred in the text then the number before that is referring to the amount awarded for special damages. To perform pattern matching a powerful technique known as regular expressions (“regex”) was used. “Regex is a string of text that allows you to create patterns that help match, locate, and manage text.”[2] . For example if we want to identify the amount in the previous example this regex “\$[0-9]+?,[0-9]{3,}” will pull out $20,000. The expression will match any string of characters that begins with a “$”, which is followed by one or more digits, which is followed by a “,”, which is then finally followed by three or more digits. The problem with this approach is that legal cases are not so simple and it is difficult to engineer patterns that do not over or under capture the information that needs to be extracted.

The second approach used in this paper is the classification based approach. Classification is simply the process of predicting the class or category of given data. With a classification approach a model has to be built where the model follows an algorithm to categorize and classify text. An example of

[2] Regex Definition

an algorithm that is used could be a decision tree[3], where a “tree” is drawn upside down with its root at the top which represents a condition, called nodes, based on which the tree splits into branches, called edges until it can’t split into any more categories. Each node will contain a condition that can be used to differentiate different classes.

To be able to do so we feed a classification algorithm a set of labeled data, called training data, which is data where we already know what the amount of damages and what amount of liability was apportioned to the plaintiff, if any. In other words the training data comes with an answer key for the classifier to learn from. The classifier “fits” the training data, meaning it learns the patterns that differentiate the different classes. After fitting the classifier is now ready to classify text where it does not know the answers.

Before finalizing the classifier one needs to know how well the classifier is working to fine tune it to maximize performance, therefore, a portion of the training data has been set aside before fitting the model, which is called a development set. We strip the answers before feeding the development set to the classifier to see to what degree it is able to correctly predict the answers. Fine tuning is performed after inspecting the accuracy and ends once the performance of the classifier is at a maximum. Since the development set is used to fine tune the classifier there is a concern that the fine tuning has forced our classifier to be very good at predicting on our development data but may not be as good at working with new data in the wild. To combat this, a third test set is also set aside from the training data which is used once at the end as a final evaluation of the classifier. Since the test set was not used to fine tune the model this final accuracy can be seen as a measure of how the classifier will truly perform. To summarize, the training data is split into a train set, development set, and test set where the classifier learns from the train set, is fine tuned using the development set, and is evaluated on the test set.

The main classification algorithms that are used in this paper are Random forest, XGBoost, Logistic regression[4]. To understand terminology in more detail refer to appendix 2.

Data

Data Source

Our primary data source for legal documents was LexisNexis. LexisNexis is a corporation, headquartered in New York City, which specializes in computer-assisted legal research[5]. They make large amounts of legal cases easily accessible electronically. All cases that were pulled from LexisNexus were meant to be negligence cases in BC and were queried from the website using the following search parameters.

- Limit to cases identified as torts
- Limit to cases between January 1st, 2000 and April 19th, 2020
- Limit to cases where the word “negligence” appears

[3] Decision Tree
[4] Random forest, XGBoost, Logistic regression
[5] Lexis Nexis - About Us

From the initial query, we pulled a total of 4,118 cases from the database. The data came formatted as 85 individual Microsoft Word .DOCX formatted files. The data was converted from the original .DOCX format into a more lightweight .TXT format. It is unnecessary to keep additional information from the .DOCX format as we are primarily interested in parsing through the actual text of each document to pull out relevant information. One major limitation in our approach is that we are unable to make any use of information that is stored in images, such as a table of damages inserted into the .DOCX file as an image rather than text. In general, a majority of cases followed the same structured format; beginning with information such as the case title, entities involved, hearing date, judge, decision length, case summary, and location of the hearing. Due to the structured nature of this information it allowed for easy extraction using a combination document line numbers and patterns.

Data Filtering & Preprocessing

Unfortunately, not every case in our collection of 4,118 was relevant to our analysis. Time had to be spent filtering out cases that would not award any damages to improve the reliability of our analysis. The full list of cases filtered out of our analysis is found below.

Criteria	Description	Number of Cases
“R. v. *Defendant*” in case title	“R. v.” signifies a crown case involving criminal law	43
Any case without “British Columbia Judgements”	If the originating reporter is not B.C.J. we exclude the case to boost the reliability of our study of cases exclusively in BC	190
“(Re)” in case title	Any case involving “(Re)” will not share the common plaintiff-defendant dynamic and will fall under interlocutory matters. In other words, they will not have a final decision made by the judge.	15
Client & Solicitor cases	Does not follow the common plaintiff-defendant dynamic.	17
“In the matter of” instead of “Between plaintiff & defendant”	Does not follow the common plaintiff-defendant dynamic.	12
Third Party Procedure	Deals with the third party of a case. Typically will involve an argument for removing or adding a third party. Does not award damages.	60
Disposition without Trial	Deals with the defendant asking the judge to dismiss a case based on a lack of evidence or proper documentation from the plaintiff.	73
Applications and Motions	Deals with applications or motions brought up with the judge. Will not award any final damages.	20
Right to a Jury	Deals with applications requesting case to be heard only by a judge without a jury	10

Adding or Substituting Parties	Similar to third party procedure. Usually deals with applications requesting to add or remove a defendant	20

Table 1 - Case Filtering

The table above will sum up to 460 cases, 15 of which overlap, causing us to remove a total of 445 cases out of 4,118. This resulted in our final set to include 3,673 cases for analysis.

Case Format

Each case pulled from LexisNexus follows a very similar structure. The beginning of each case includes information like the case title, judge name, registry location, decision length, case summary, and the plaintiffs, defendants, and third parties involved in the case. Within the case summary, the report may include the result of the case in the form of “HELD: Application Dismissed” which allows for filtering of cases where the plaintiff lost. Sometimes the summary may include damages awarded or liability apportioned which we are attempting to extract. The case report will follow the case summary where each line begins with a paragraph number unless it is a section header which is used to organize the report into an easier to read format. Section headers are not standard across cases but some common headers will be used such as “Conclusion”, “Summary”, or “Damages”. When converting the cases into a TXT format, the headers were surrounded by XML tags which are used in our classifier based approach. Surrounding the header with a specific tag will allow the classifier to be able to understand which section it is currently inspecting; headers are useful for both humans and machines because they give additional context about the text. Typically, the information we are trying to extract will appear either in the case summary or near the end of the case in the conclusion. Finally, each case will end with an “End of Document” marker. An example of a typical case that we need to extract information from is included in appendix 3.

Annotations

In our methods section we discuss the use of annotated training data for use in the classification based approach. In general, training data can be thought of as data that a machine learning algorithm uses to learn from. The algorithm uses knowledge from examples it has already seen and applies it to new examples that it has not seen before. The idea is that we can manually create a training set from a small sub sample of cases which the algorithm can use to provide useful information on the entire set of cases. The information that we wish to feed our classifier would be any numeric value found in a case and the result we want from the classifier is what type of damage, if any, is it. Therefore, in our training data we must annotate every numerical value and assign a class or category to it. One method of doing so is to insert tags around the values directly in the .TXT file which can be easily read by a computer. An example sentence we wish to annotate may be,

“I award the plaintiff $20,000 in special damages”

This would be annotated manually to,

“I award the plaintiff <damage type = ‘special’>$20,000</damage> in special damages”

We would annotate liability (contributory negligence) in a similar fashion and use the word "percentage" rather than "damage" because liability is apportioned as a percentage fault towards the plaintiff or defendant. We manually went through 298 cases and annotated each numeric dollar value or percent value. The possible classes we could have assigned to a value is included below.

Tag Name	Tag Type
Damage	"Other" - Any value that is not a final damage (e.g. plaintiff is seeking this value or the plaintiff is referencing a damage payout from a previous case) "Non pecuniary" "Special" "In Trust" "Past Wage Loss" "Future Wage Loss" "General" "Punitive" "Aggravated" "Future Care" "Reduction by" - Another value is reduced by this amount "Reduction to" - Another value is reduced to this amount "Total" - Total damages awarded in a case "Total After" - Total damages awarded after contributory negligence
Percentage	"Other" - Any percentage that is not contributory negligence "CNP" - Contributory negligence percent for the plaintiff "CND" - Contributory negligence percent for the defendant

Table 2 - Annotation tags

Overall, the tags presented above were able to cover the different types of information we needed to extract for our analysis. However, our biggest challenge is when a case has damages broken down into an itemized list. For example, the court may award a plaintiff money for past wage loss at different jobs and may not sum the values. In this case we would prepend the term "sub-" in front of the tag, leading to the tag becoming "sub-past wage loss", to indicate that the value is not the final value but is a partial value. The intuition behind this is that our classifier based approach may be able to learn the difference between certain damages being broken down into individual items compared to the final sum of a damage type. An example of a case annotation can be seen in appendix 4.

Data Statistics

As stated earlier, our data ranges from the years 2000 to 2020. The distribution of cases per year is relatively even with the exception of 2020 due to the fact that we pulled the case information part way through the year.

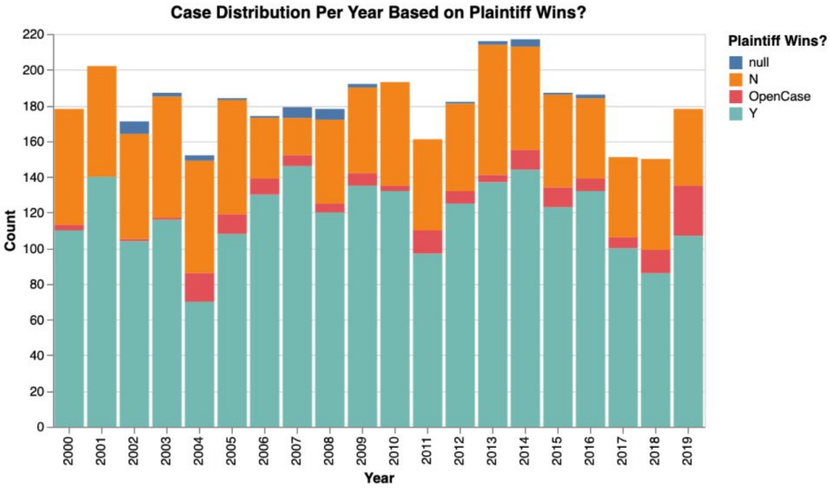


Figure 1 - Case Distribution per Year Based on plaintiff Winning or Not
Y(green) suggests plaintiff winning, N(orange) indicates plaintiff loosing, OpenCase(Red) indicate undecided cases and null represents the cases classifier could not decide whether won or not

We tagged 5,168 individual numbers for the damage tags and 997 individual percentages for the percentage tags. Of the 5,168 damage tags there were 3,006 (58.2%) with the tag type of "other". This shows that the majority of values mentioned in a case will not be the amount that is awarded. Similarly, of the 997 percentage tags we annotated there were 695 (69.7%) with the tag type other, which shows a similar observation. One different reason for percentage having so many "other" tags is because we are only tagging percentages that involve contributory negligence. The case may contain other percent reductions which would be tagged as "other". Below we show the overall distribution of each other tag type excluding the "other" tag to make a clearer visualization.

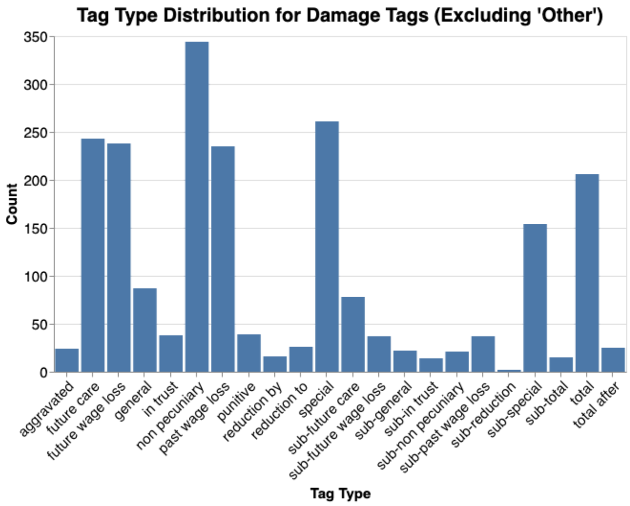


Figure 2 - Damage Tag Type Distribution

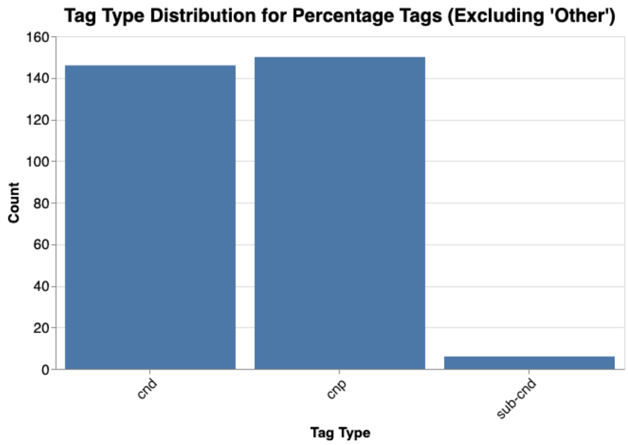


Figure 3 - Percentage Tag Type Distribution

After tagging each case we place the relevant information into a .CSV format file as well. This file stores information about the case such as the case title, judge name, location, damage awards broken down into subcategories, and contributory negligence percentages. This .CSV file can later be used to evaluate how well we are doing with our rule based approach and classifier based approach. For each of our approaches the goal is to extract all relevant information from unannotated data and export the information into a .CSV file for further analysis.

Research Questions

The main goal of this project was to answer the following research questions:

- *How have the negligence payouts changed from the year 2000 to 2019 in BC?*
- *What percentage of cases have contributory negligence and what percentage were successful?*

To answer these questions the following types of information were extracted from the text files (see method section).

String type	Number type	Boolean type
Case Number	Total Damage	Written Decision
Case Name	Total Pecuniary	plaintiff Wins
Year	Non Pecuniary	Multiple Defendants
Judge name	General	Contributory Negligence Raised
Registry	Special	Contributory Negligence Successful
	Punitive	
	Aggravated	
	Future Care	
	Decision Length of a case	
	Percent Reduction	

In addition other questions were answered using statistics for each of the damage amounts awarded based on each of the extracted fields (per year, per judge, per location, and if multiple defendants are present) please see the analysis section.

Methods

Preprocessing Word Documents

To approach this problem, we first converted all .DOCX files queried from Lexis Nexis to .txt files for ease of manipulation in Python. We used a Python library called `docx` to iteratively extract the all text and any tabulated information found in the original .DOCX files, and write the extracted text to a blank .TXT file. We found a few cases that had images embedded in the original document. These cases were rare and we chose to skip any images found.

Rule-Based Information Extraction

After spending some time understanding the common structure of all British Columbia Judgements (BCJ) negligence cases from Lexis Nexis, we decided to make a first pass by extracting desired fields per case using regular expressions (regex). The beginning of each case starts with the case name, such as *Mawani v. Pitcairn, [2012] B.C.J. No. 1819*, and the end of each case is marked by the line "End of Document". This made it easy to split documents by case. Most of the desired descriptive fields such as Case Name, plaintiff Wins, Judge Name, Multiple Defendants, Decision Length, and Registry, could also be encountered within the first 20 lines of each case. We also filtered for BCJ cases using the first few lines of each case.

Rule-Based Damage Extraction

The rules used to extract the damages awarded to each case required reading through many cases to understand common patterns. The most deciding patterns we encountered include the location in the document where final values are mentioned and the language used to describe each value. As one would expect, final values awarded by the Judge are typically mentioned near the end of each case, and reiterated at the beginning of the text under a header called *Case Summary*. Although some cases separate final decisions under a header with the word *Conclusion* or *Damages*, this was not consistent enough to use as a rule.

As shown in Appendix 1, some damage types are the sum of other damage types. For example, General damages are made up of Future Wage Loss, and Future Care, and Special Damages include In-Trust awards, Housekeeping, and Past Wage Loss. Although some cases include a cumulative value per category (eg. Total, Non-Pecuniary, General, Special, etc), many do not. This added another layer of complexity to our rule-based methods for extracting damages.

Ultimately, the rules used to extract damages include the context around dollar amounts (up to 10 tokens), the ratio of paragraph number where the value was found to the decision length, and vocabulary specific to each damage type. First, we used a regex pattern to split the case into numbered paragraphs. Next we searched each paragraph for a dollar amount. If the paragraph location ratio is greater than 0.9 or the value was found in the *Case Summar*y section, we continue to apply rules to categorize values, otherwise, we disregard the dollar value match. If the value is located in the last 10% of paragraphs or in the summary, we then compared the context surrounding the value to the common language specific to each category to determine which damage type, if any, best fits the value. If no Total damage value was found in the text, we added Non-Pecuniary and Pecuniary damages to get a total.

Rule-Based Contributory Negligence Extraction

The methods used to extract the percent reduction [of damages] due to contributory negligence are similar in nature to those used to extract damages. Before seeking individual percentages mentioned in the case, we filtered for cases in which the plaintiff won and the phrase *contributory negligence* was mentioned somewhere in the text. After filtering, again, we started by iterating through numbered paragraphs in the case and assigning a score to each paragraph based on where it occurred in the text. If the paragraph score was greater than 0.6 , we continued to apply conditions to determine if the percentage was related to contributory negligence, otherwise we continued to the next paragraph. The paragraph score of 0.6 is a ratio that we tuned to ensure the rule-based method was not over or under assigning the percentage as contributory negligence.

For any percentages found whose location scored above 0.6, we searched up to 8 tokens on either side to check for specific vocabulary related to contributory negligence. An example of some of this vocabulary includes the following words: contributory, liability, apportion(ed), fault, against, recover, and responsible. Depending on the language, the percent fault is either mentioned against the plaintiff or against the defendant. For this reason, another rule was utilized to determine whether the word *plaintiff* or the plaintiff's name (extracted from the case name) was found in the context or the opposite was true for the defendant. If a percent was found located in either the last 40 percent of paragraphs with the appropriate language in the context, a final check was made to ensure either the word *plaintiff*, *defendant,* or *damages* was also found in the context. If so, it was determined that the field *Contributory Negligence Successful?* was true, otherwise it was deemed false. If contributory negligence was not found to be successful after searching through all paragraphs in the case, the same process was repeated to check percentages in the Case Summary.

Rule-Based Evaluation

To test how well our rule-based information extraction was performing, we created a function that computes accuracy of empty fields, filled fields, and overall accuracy across each attribute we extracted. These accuracies all compare the values extracted by our rule-based methods to the true values annotated by a combination of the three of us. Empty accuracy is defined as the number of times our rule-based extraction correctly determined a field was empty, out of the total number of empty values for that attribute. Filled accuracy is the ratio of correctly identified values in a certain field out of the total number of filled values for the same field. For example, many cases do not have damages awarded - these cases would have an empty entry for all damage quantities. If our rule-based method correctly identifies that the case does not have any damages, this is a correct empty field. Below in table 3 are the tabulated rule-based accuracies:

Columns	Overall accuracy (%)	Filled field accuracy	Empty field accuracy
Case Name	100	100	NA
$ Damages total	50.35	29.38	97.67
$ Pecuniary Damages Total	26.07	17.28	83.78
$ Non-Pecuniary Damages	82.14	53.27	100
$ General Damages	77.14	44.21	94.05
$ Special damages	80.71	51.81	99.41
$ Punitive Damages	98.21	60	99.62
$Aggravated Damages	98.92	66.66	100
$Future Care Costs	87.85	59.74	98.52
Decision Length	99.28	99.28	0
Multiple Defendants?	96.07	96.07	0
plaintiff Wins?	86.42	86.42	0
Registry	97.28	97.85	0
Cont. Neg. Success?	93.57	93.57	0
% Reduction	93.57	84.09	95.33

TABLE 3: Rule-Based Information Extraction Accuracies

As shown in table 3, our rule-based information extraction does quite well for both filled and empty fields for the fields such as Registry, plaintiff Wins, Multiple Defendants?, Decision Length, and Case Name. As anticipated, these values are stored in a more consistent format, and therefore results are upwards of 85% accuracy for both filled and empty fields. Accuracies for damages awarded performs quite well when there are not any damages awarded (empty fields), but does not do so well for filled values. There is such variation in the structure for how these values are stored, as well as many irrelevant dollar values mentioned, that it is hard to make enough rules to correctly capture the values. The performance of *Contributory Negligence Successful?* And *% Reduction* performed a lot better than anticipated after fine-tuning the rules and optimal paragraph location ratio cut-off.

To improve results for damages extracted, we built a classifier to categorize dollar values as described in the *Annotations* section above.

Classification

Damages Classifier

Given the class imbalance between tagged damage types as described in the *Data Statistics* section above, the low volume of annotated data, and the need to differentiate between the 23 classes being predicted, and large feature set inherent to text classification, we decided to compare results across 3 classifiers: Logistic Regression, Random Forest, and XGBoost. Both Logistic Regression and Random Forest were implemented using the `sklearn` packages, where XGBoost uses the open-source library[6]. First we preprocessed the text in each case by lowercasing all words and removing stop words included in nltk.stopwords[7] (*and*, *the, if, of, an, to, as, but, their* etc). The pipeline for building the training set for these classifiers is as follows:

1) For each annotated case, using a regex iterator, loop through each tagged damage
2) For each tagged damage:
    a) extract the quantity tagged (damage value) and tag type (true label)
    b) extract context around the value and build list of dictionaries, where each dictionary is a feature, based on word counts in the context (Bag-of-Words) among other features
3) Use the sklearn DictVectorizer[8] to turn the list of dictionaries into a matrix of shape number of tagged examples by number of features

As enumerated in the pipeline above, to implement these classifiers, we built two functions that iterate through all annotated cases, use a regular expression to find tagged damage values, and extract several fine-tuned features into a list of dictionaries for each tagged value. The feature engineering process is described in detail below.

Feature Engineering

To start, we chose many of the same features that were used in the rule-based method for damage extraction. The primary feature includes a Bag-of-Words (BOW) representation of the context surrounding each value. BOW maps each word in the context to the number of times it appears. This allows the classifier to associate certain words with each category. In addition to the BOW counts of each word in the text, we added a BOW feature that has word counts for where the word occurs relative to the value, before or after. Another feature adopted from the rule-based method is what we call *start_idx_ratio* which represents the location of the value relative to the start of the text. Values near the end will have a ratio close to one and values near the start have a ratio close to zero. In addition to the location and BOW features, we included a range in which the value itself falls. These bucket ranges include: values less than $1,000, values between $1,000 and $25,000, values from $25,000 - 100,000, $100,000-500,000, and values above $500,000. For example, a value of $5,000 falls in the bucketed range of $1,000 and $25,000. Another feature we added is based on the header above the dollar value of interest. Often, cases have headings such as *Future Care Awards, Wage Loss, Summary of Damages,* etc. An example of a feature set for a single value is shown below:

```
{'case@Heading': 1,
'summary@Heading': 1,
 'start_idx_ratio': 0.04429809982826436,
'range': '100000 - 500000',
```

[6] Open Source Library
[7] nltk.stopwords
[8] sklearn DictVectorizer

'prev word': 'services,',
'next word': 'accommodations',
'prev bigram': 'therapies services,',
'next bigram': 'accommodations renovations,',
'prev trigram': 'equipment, therapies services,',
'next trigram': 'accommodations renovations, $175,000',
'age@Before': 1, '19,@Before': 1,
'$950,000@Before': 1,'Equipment,@Before': 1,
'therapies@Before': 1,'services,@Before': 1,
'age': 1, '19,': 1, '$950,000': 1,
'equipment,': 1, 'therapies': 1,
'services,': 1, 'accommodations': 1,
'renovations,': 1, '$175,000': 1,
'in-trust': 1, 'past': 1, 'future': 1}

Ablation Study

To optimize our set of features, we used a process called ablation which involves removing one feature at a time and comparing precision and F-score results before and after. In Table 4 below, the ablation results are shown for each classifier tested. In our final feature set, we used the least number of features resulting in highest precision performance for our best classifier, shown in the last row.

Feature Removed	**Overall Precision (weighted)**	**Macro F-Score(weighted')**
ALL Features	0.68959	0.6483
Float Bins	0.66183	0.6303
start_idx_ratio	0.68491	0.6471
prev word	0.68062	0.6449
next word	0.68000	0.6408
next bigram	0.68126	0.6412
next bigram and prev bigram	0.67756	0.6415
prev/next bigram and trigram	0.66654	0.6338
prev/next word, bigram and trigram	0.61969	0.5918
BOW before	0.68646	0.6423
BOW after	0.68959	0.6521
BOW (unordered)	0.68958	0.6383
headers	0.67706	0.6289

Table 4: Feature Ablation

Note: The following features were ignored in the weighted calculation - *other, total after,* all *reduction* categories

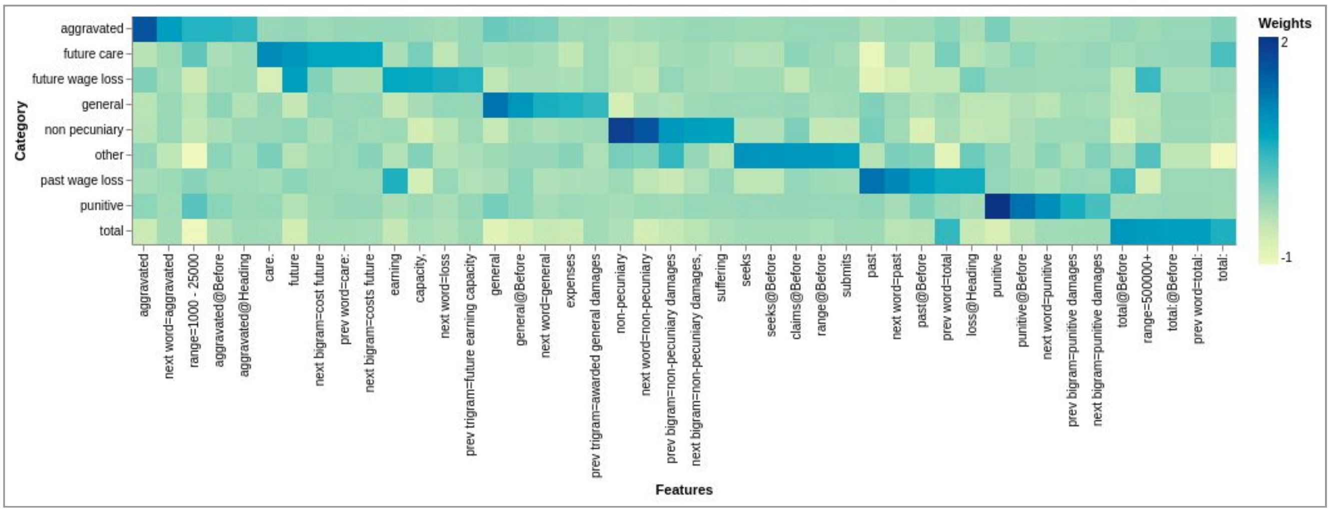


Figure 4: Top 5 Features Per Category: Feature Weights

As shown in Figure 4 above, our model was able to pick out sensible features for predicting damage types. For example, features for the category *future care* include the words *care, future,* bigram *costs future,* and previous token equal to *care:.* The feature *future* has shared importance for the other damage type *future wage loss*. One of the best examples that our classifier is working well, are the top five features for the category *other*. Namely: *seeks, seeks@Before, claims@Before, range@Before,* and *submits*. Often the plaintiff "seeks" a damage award of some value, but it is ultimately not what we want to pick out of the text.

Hyper-Parameter Optimization

Once we determined the best combination of features across all classifiers, we ran a cross-validated grid search using the *sklearn* package GridSearchCV[9] to optimize for the best subset of classifier hyperparameters. *GridSearchCV* compares the performance of the classifier specified using all combinations of parameters specified. While Random Forest and XGBoost have few possible hyperparameters to tune, Logistic Regression has several. To control for randomness, we set the `random_state` parameter in each classifier to 42 throughout our research. For Logistic Regression, we tested for different combinations of the following hyperparameters: {'penalty':['l1', 'l2'], 'C':[0.1, 1, 10, 100], 'solver':['liblinear', 'newton-cg'], 'max_iter':[600], 'class_weight':[None, 'balanced']}. The best results for Logistic Regression to classify damages include: {'C': 1.0, 'class_weight': None, 'max_iter': 600, 'penalty': 'l2', 'solver': 'newton-cg'}. The two tree-based classifiers performed best using default values.

Damages Classifier Evaluation

After feature engineering and hyperparameter tuning, the final performance across the three damages classifiers tested can be found below in Table 4.

[9] GridSearchCV

Classifier	Overall Precision (weighted)	Overall F-Score (weighted)
Logistic Regression	0.68959	0.6483
Random Forest	0.72247	0.5888
XGBoost	0.69038	0.6116

Table 4: Final Results for comparing classifiers

Although all three classifiers performed similarly, we wanted a high precision model across all categories, but especially for Total Damages.
We decided to use the Logistic Regression model for final analysis both because of its high precision performance, but also because of its `predict_proba`[10] method which provides a probability distribution across all categories for each value. This is useful for selecting high-confidence results for unseen data.
The overall performance of the optimized Logistic Regression model is shown in Table 5 below:

category	precision	recall	f1-score	support
aggravated	0.93	0.62	0.74	21
future care	0.74	0.78	0.76	241
future wage loss	0.71	0.72	0.71	237
general	0.69	0.53	0.6	87
intrust	0.6	0.32	0.41	38
non pecuniary	0.8	0.85	0.83	343
other	0.87	0.95	0.91	2980
past wage loss	0.73	0.73	0.73	231
punitive	0.9	0.9	0.9	39
special	0.74	0.8	0.77	256
sub-future care	0.53	0.13	0.21	78
sub-future wage loss	0.5	0.19	0.27	37
sub-general	0.33	0.09	0.14	23
sub-intrust	0.83	0.36	0.5	14
sub-non pecuniary	0.73	0.38	0.5	21
sub-past wage loss	0.00	0.00	0.00	37
sub-special	0.46	0.2	0.28	153
sub-total	0.86	0.4	0.55	15
total	0.67	0.66	0.66	203
total after	0.71	0.2	0.31	25
accuracy	NA	NA	0.81	5123
macro avg	0.65	0.46	0.51	5123
weighted avg	0.79	0.81	0.8	5123

Table 5: Classification Report - Best Results, Damage Classifier

[10] Predict_proba

Applying Damages Classifier to Unseen Data

Once we determined the optimal feature set and Logistic Regression model, we implemented two functions to apply the model to unseen data: one that predicts dollar values found in the text, and another that assigns a final prediction for each damage type to each unseen case. Given an unseen negligence case, we then use the optimized model fit on all annotated data, and the `sklearn` vectorizer used to transform the annotated features to predict the damage category. The `predict` function follows this general procedure:

1) Iterate over all dollar amounts found in the case using a regex iterator
2) Extract features for the value found and append to a list
3) Store the float value of the dollar amount found
4) Once steps 1-3 are repeated for all values, transform the features using the fit-transformed vectorizer
5) Use the fit classifier to predict the damage tag and associated probability distribution

Using the results obtained from our `predict` function, we then pass results into the `assign_classification_damages` function which follows this pipeline:

1) Iterate over all predictions and associated probabilities in a single negligence case
2) If the maximum probability (between 0 and 1) associated with the predicted tag is above the threshold we set (0.5), add the tag to a dictionary mapping the predicted tag to a list of possible high-scoring values
   a) {general:[5000, 5000, 3000, 5000], special: [10000, 7000], non pecuniary: [80000, 80000], ...}
3) Once all predictions have been assigned to the dictionary of possible values per category:
   a) If the tag is of type *other*, continue
   b) If the keyword argument *high_precision_model* is:
      i) *True*, skip all 'sub' categories and choose the last value present in each list of possible values
      ii) Otherwise, if the category is of type 'sub' (eg *sub-special* or *sub-general*) and there are no non-sub versions of the same tag (eg *special* or *general*), then we add the values in the list
   c) If no values were found in either a *sub* category or a non-sub category, the final value is set to *None*
   d) Assign the final value per damage type to a dictionary
4) Return the final damage assignments

We modified our original parsing function that uses our rule-based methods as default, to allow for use of a classifier to assign values for damages and contributory negligence percentages. The output of this parsing function contains a list of dictionaries where each dictionary contains the values extracted for a particular case.

Contributory Negligence Percent Reduction Classifier

Feature Engineering

The feature engineering process for our percent reduction classifier was quite similar to that used for the damages classifier. We started using a BOW representation and differentiated word counts for words that appear before and after the percent value of interest. We also added a float equivalent of the tagged string percentage as well as the start index ratio feature telling where the value lies relative to the end of the case. We also made our own contributory negligence-related lexicons containing words such as *fault, against, apportion, responsible, recover,* and *contributorily*. The one feature that we used for percent reduction that we did not need to apply for damage quantities was the presence of the word *plaintiff* or *defendant* and their respective names. As described in the rule-based methods for percent reduction, we used a regex to extract the two names from the case title. This proved to improve accuracy by a few percent as shown in Table 6 below tabulating our ablation study comparing the effects of each feature on model performance. Removing the *start_idx_ratio,* and *lemmatization* proved to have the best impact of classification performance.

Feature Removed	**Overall Precision (weighted)**	**Overall F-Score (weighted)**	**Overall Accuracy**
ALL Features	0.62191	0.6281	0.81
Float	0.60723	0.61337	0.81
Float and Bins	0.56792	0.56924	0.8
Bins	0.60716	0.60845	0.81
start_idx_ratio	0.62680	0.63294	0.82
defendant name (Boolean)	0.62191	0.62805	0.81
plaintiff name	0.62191	0.62805	0.81
plaintiff and reduction	0.59736	0.59865	0.8
defendant and reduction	0.61208	0.60845	0.81
reduce lexicon	0.62683	0.61828	0.81
defendant mentioned	0.61211	0.61825	0.81
plaintiff mentioned	0.62191	0.62805	0.81
bow (unordered)	0.54921	0.55630	0.77
bow after	0.62191	0.63789	0.82
bow before	0.60719	0.62317	0.81
lemmatization	0.62686	0.63300	0.82
lemmatization, start_idx_ratio, BOW_a, reduce_lexicon	0.64152	0.62805	0.82
lemmatize, start_idx_ratio	0.64155	0.64769	0.83

Table 6: Feature Ablation Results, Not Including ‘Other’ category

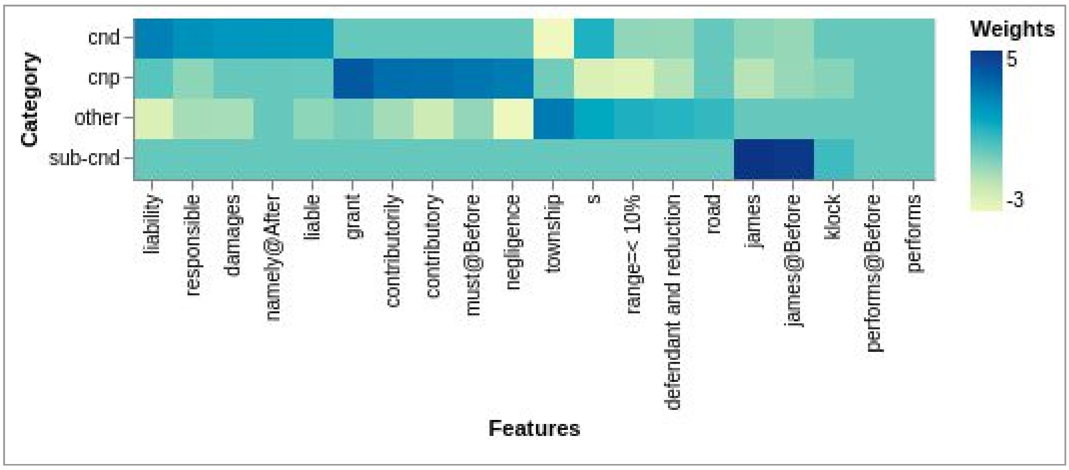


Figure 5: Top 5 Most Important Features, Percent Reduction Classifier

As shown in Figure 5 above, the most important features used to predict our contributory negligence percentages against the defendant (*cnd*) include words such as *liability, responsible,* and *damages*. Features used to predict contributory negligence percentages against the plaintiff (*cnp*) include: *grant, contributorily, contributory,* and *negligence*. You can tell by the feature weights that our model is less sure about predicting the *other* category of percentages, though the feature *range<10%* is a good indication; our team has not seen a percent reduction value less than 10% mentioned in a case during the annotation process. Finally, the *sub-cnd* category is quite certain about its predictions (dark blue) given the features *james*, and *james@Before*. This is indicative of overfitting. We only have 6 examples of annotated *sub-cnd* features from a single case, in which James is one of the defendants. For this reason, we did not use the *sub-cnd* category to assign percent reductions for evaluation.

Hyper-Parameter Optimization

The same methods were used to optimize our model parameters for the contributory negligence percent classifier as those for the damages classifier. For more details,please refer to that section above.

Contributory Negligence Classifier Evaluation

After feature engineering and hyperparameter tuning, the final performance across the three percent reduction classifiers tested can be found below in Table 6 :

Classifier	Overall Precision (weighted)	Overall F-Score (weighted)	Overall Accuracy
Logistic Regression	0.64155	0.64769	0.83
Random Forest	0.62766	0.58195	0.8
XGBoost	0.65710	0.56617	0.81

Table 6: Comparing Performance Across Percent Reduction Classifiers

For the same reasons discussed in the *Damages Classifier Evaluation* section, we decided to move forward with the Logistic Regression classifier for our analysis. Below in Table 7 is a more detailed chart of the optimized Logistic Regression results:

category	**precision**	**recall**	**f1-score**
Cnd	0.69	0.72	0.71
cnp	0.59	0.56	0.58
other	0.9	0.9	0.9
sub-cnd	0.71	0.83	0.77
accuracy	NA	NA	0.83
macro avg	0.72	0.76	0.74
weighted avg	0.82	0.83	0.82

Table 7: Classification Report - Percent Reduction Classifier

Applying Contributory Negligence Classifier to Unseen Data

Similar to applying the damages classifier to unseen negligence cases, once we determined the optimal feature set and optimized the Logistic Regression model, we implemented two functions to apply the model to unseen data: one that predicts percentages found in the text, and another that assigns a final prediction for each percentage type to each unseen case. As described in the *Annotations* section above, our percent reduction classifier distinguishes between 3 main classes: *CNP* which is percent reduction against the plaintiff, *CND* which is percent reduction against the defendant, and *Other* for percent values which are unrelated to contributory negligence. Given an unseen negligence case, we then use the optimized model fit on all annotated data, and the *sklearn* vectorizer used to transform the annotated features to predict the percent category. The *predict* function follows this general procedure:

6) Iterate over all percentage amounts found in the case using a regex iterator
7) Extract features for the percent found and append to a list
8) Store the float equivalent of the percent found
9) Once steps 1-3 are repeated for all values, transform the features using the fit-transformed vectorizer
10) Use the fit classifier to predict the damage tag and associated probability distribution

Note that because the *predict* function using the damages classifier is almost identical to that for the percent reduction classifier, we added a few keyword arguments to use the same function for both.

Using the results obtained from our *predict* function (with the appropriate keyword argument telling it which regex pattern to use), we then pass results into the *assign_classification_CN* function which follows this pipeline:

5) Iterate over all predictions and associated probabilities in a single negligence case

6) If the maximum probability (between 0 and 1) associated with the predicted tag is above the threshold we set (0.7), add the tag to a dictionary mapping the predicted tag to a list of tuples of possible high-scoring values, along with their associated probability. For an example:
    a) {CNP:[(0.5, 0.88),(0.3, 0.97), (0.5, 0.71)], CND: [(0.7, 0.88)]}
7) Once all predictions have been assigned to the dictionary of possible values per category:
    a) If the tag is of type *other*, continue
    b) If the category is of type *CND*, we need to subtract the float from 1 because the percent is against the defendant, and we are interested in the equivalent value against the plaintiff
    c) Otherwise, choose the most probable value from the list of tuples, (value, probability)
    d) If no values were found in either a *CND* category or *CNP* category, the final value is set to None
8) Return the final, most probable percentage for the case

Overall Classification Evaluation

To perform an overall analysis of our classification results, similar to the rule-based evaluation, we compared filled and empty value accuracies between the final assigned predictions for both damages and percentages, against the gold annotated data set.

Columns	Overall accuracy (%)	Filled field accuracy	Empty field accuracy
$ Damages total	43.75	28.16	98.0
$ Pecuniary Damages Total	26.34	19.8	74.07
$ Non-Pecuniary Damages	88.84	94.26	82.35
$ General Damages	76.34	46.67	96.27
$ Special damages	76.79	55.24	95.8
$ Punitive Damages	99.11	90.0	99.53
$Aggravated Damages	100	44.44	97.77
$Future Care Costs	90.18	76.71	96.67
% Reduction	92.86	87.8	93.99

Table 8: Overall Accuracies Using Classification

High Precision Classification Results - Damages

Columns	Overall Accuracy (%)	Filled Accuracy (%)	Empty Accuracy (%)
$ Damages total	69.44	70.4	0
$ Non-Pecuniary Damages	87.5	85.11	92
$ General Damages	69.44	54.35	96.15
$ Special damages	68.06	58.49	94.74

$ Punitive Damages	98.61	66.66	100
$Aggravated Damages	98.611	66.66	100
$Future Care Costs	77.77	69.23	87.87

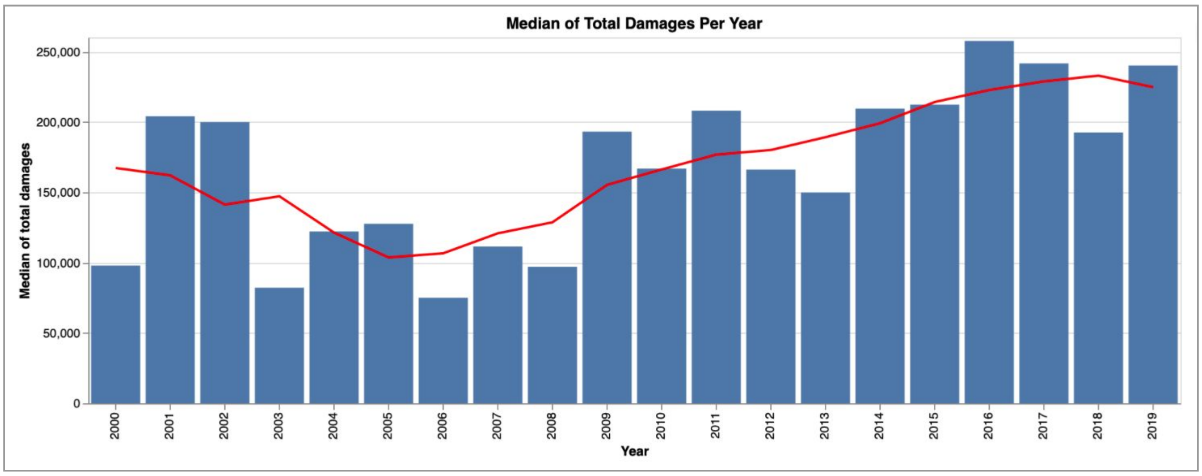
Table 9: High Precision Damages

As mentioned in the *Applying Damages Classifier to Unseen Data* section above, we set a probability threshold that the predicted damage must meet in order to be assigned. In order to achieve high precision results, we added a keyword argument called *high_precision_mode*. If *high_precision_mode* is *True*, we only use predicted tags found in the text and do not add any sub-categories together. Adding sub-categories has shown to propagate errors leading to lower filled accuracies. For example, if a *total* damage value is not found in the text, instead of adding the *pecuniary* and *non-pecuniary* damage results to get a total, we ignore the *total* field (set it to *None*). As shown in Table 9 above, this is shown to result in more accurate filled values. That said, the more strict our probability threshold, the less values we use from our classifier. We modified the threshold to maximize accuracy as well as the number of values being predicted in each category, specifically the *total damages* category.

Analysis

One of the main goals of this project was to analyse BC negligence cases and see whether the trend of damages awarded increases or decreases since year 2000. Two main approaches were used to extract results from the text files, one is rule base approach and other is classification approach. The results show that classification method is the most accurate and precise approach.(Table 8)

Of the classification algorithms built to predict damages such as Decision Tree, Random Forest, XGboost, lgbm, SVC, and Logistic Regression, the latter generated the most promising and precise results with overall accuracy of 68% and F-score of 64% shown in Table 6.
The predicted results with the logistic regression classifier clearly shows that there is an increasing trend in median of total damages awarded per year after being adjusted for inflation (Graph1).

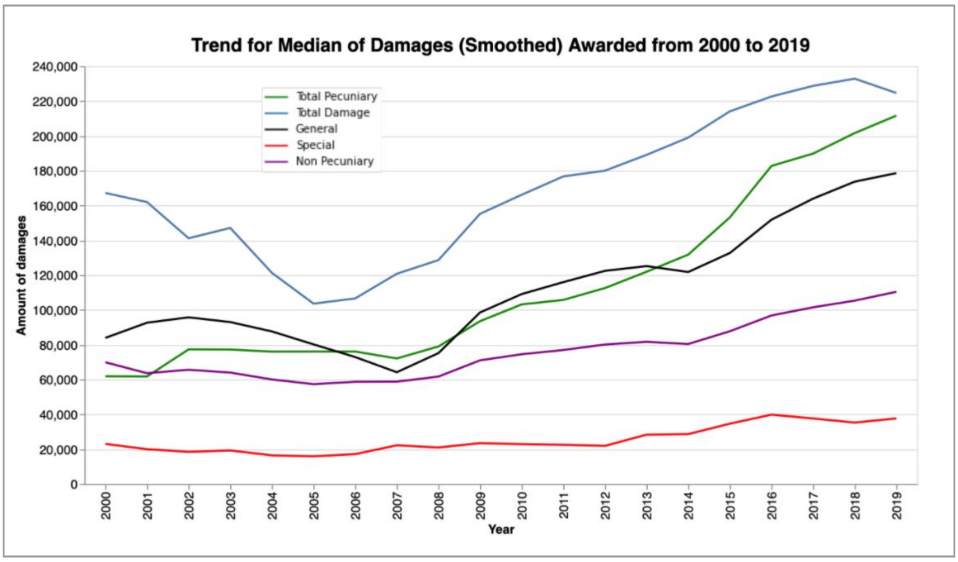


Graph1: Illustrates bar plots (blue) of median of total damages over years, layered with smoothed line plot (red) of median total damages per year.

Upon analyzing the trend of median total damage awarded per year, with 69% precision (table 5) it is evident from graph1 that the total damages awarded have been increasing over the past 20 years. To confirm that the results are not due to chance, we performed statistical analysis to test the validity of the predicted results. Pearson correlation analysis measures linear correlation between two variables X and Y. Correlation has a value between +1 and −1, where 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation.

The Pearson correlation between year and total damage awarded is 0.73 which indicates a strong positive correlation between year and damage awarded. In addition, we performed linear regression and associated p-value, a type of Inferential Statistics[11], which focuses on the strength of the relationship between two or more variables. Linear Regression analysis[12] assumes an independence between independent variables and a dependent variable. In this case the null hypothesis for regression analysis would be that the year and total damage are independent of one another therefore the slope of total damages over the years would be zero, or a flat line. Regression analysis outputs p-value[13] which is "the probability of obtaining a test statistic just as extreme or more extreme than the observed test statistic assuming the null hypothesis is true" . The p-value obtained was 0.00014 , which is smaller than the alpha value[14] of 0.05 therefore the result is significant. Significant results suggest that the null hypothesis is rejected and the slope between year and total damage awarded is not 0, so it confirms that there is a dependence between year and total damage. As a result, predicted total damages awarded using logistic regression classifier and performance of statistical analysis proves that the trend of total damage awarded in BC negligence cases has increased from year 2000 to 2019.

Trend of the median of the damages awarded for sub categories such as total pecuniary, non pecuniary, special, and general damages were also analysed when predicted by logistic regression classifier and the results showed that most types have a trend of increasing over years. Graph 2.



Graph 2: Shows a line plot for median Total, General, Non Pecuniary, Pecuniary, Special damages from year 2000 to 2019, smoothed with window of 5 (refer to method section for smoothing function).

[11] Inferential Statistics
[12] Linear Regression analysis
[13] p-value
[14] alpha value

This graph shows a decrease of total damages from 2000 to 2007 and then a steady increase until year 2019.
The trend of lines suggest that the median damages awarded are increasing per year for the non pecuniary, Total Pecuniary, General damages, and Special damages (precisions and accuracies found in table 5) . non pecuniary damage has the steepest slope amongst other sub categories. Same as before, to prove that this trend is not due to chance, some statistical analysis was performed and the results are shown in table 9.

Damage Type	Correlation	P-value	Significant?
Non Pecuniary	0.76	5.82×10^{-5}	YES
Total Pecuniary	0.79	1.72×10^{-5}	YES
General	0.59	0.0045	YES
Special	0.63	0.001	YES
Future Care	0.37	0.097	NO

Table 9: Displays correlation analysis and linear regression statistical inference p-value for damage types Non Pecuniary, Total Pecuniary, Future Care, General and Special.

Non Pecuniary damage awarded has a correlation of 76% with year and the significant p-value suggests that it’s non pecuniary damage slope is not zero (rejecting the null hypothesis of slope = 0) and it’s increasing year to year. As we can see from the predictions as well, the mean and median of non Pecuniary damage per year has an increasing trend (fig6).

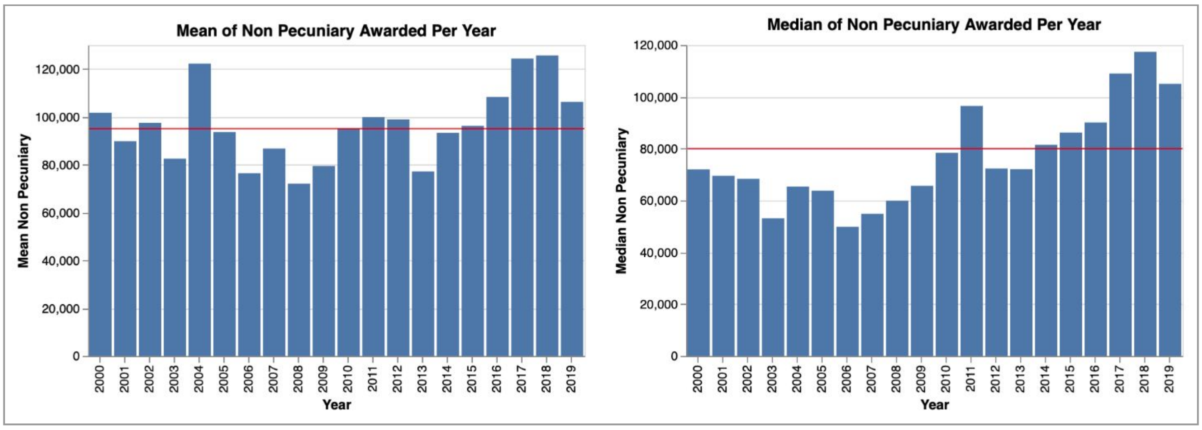


Fig 6: The mean (left), and Median (right) of non pecuniary damages awarded from year 2000 to 2019. Year on x-axis and damage amount on y_axis. Red line indicates the mean(left) and median(right) of damages over the years.

Total Pecuniary damage awarded has a correlation of 79% with year and the significant p-value suggests that it’s pecuniary damage slope is not zero and it’s increasing year to year. As we can see from the predictions as well, the mean and median of pecuniary damages per year has an increasing trend. (figure 7)

Both figure 6 and figure 7 median’s (right hand sides) shows a better visualization of the damage trend than the mean, since median neutralizes the effect of outliers and gives a more clear understanding of the trends.

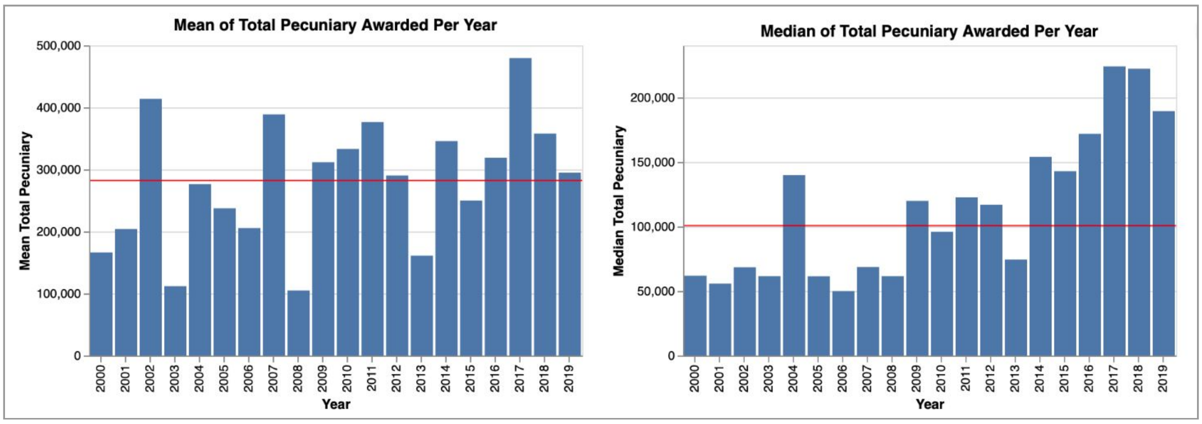


Figure 7: The mean (on the left), and Median (right) of total pecuniary damage from year 2000 to 2019. Year on x-axis and damage amount on y_axis.

General and Special damages awarded also have a positive correlation of 59% and 63% with year and the significant p-value suggests that both damages slopes are not zero and it's increasing year to year. As we can see from the predictions as well, the median of General and Special damages per year has an increasing trend (Figure 8).

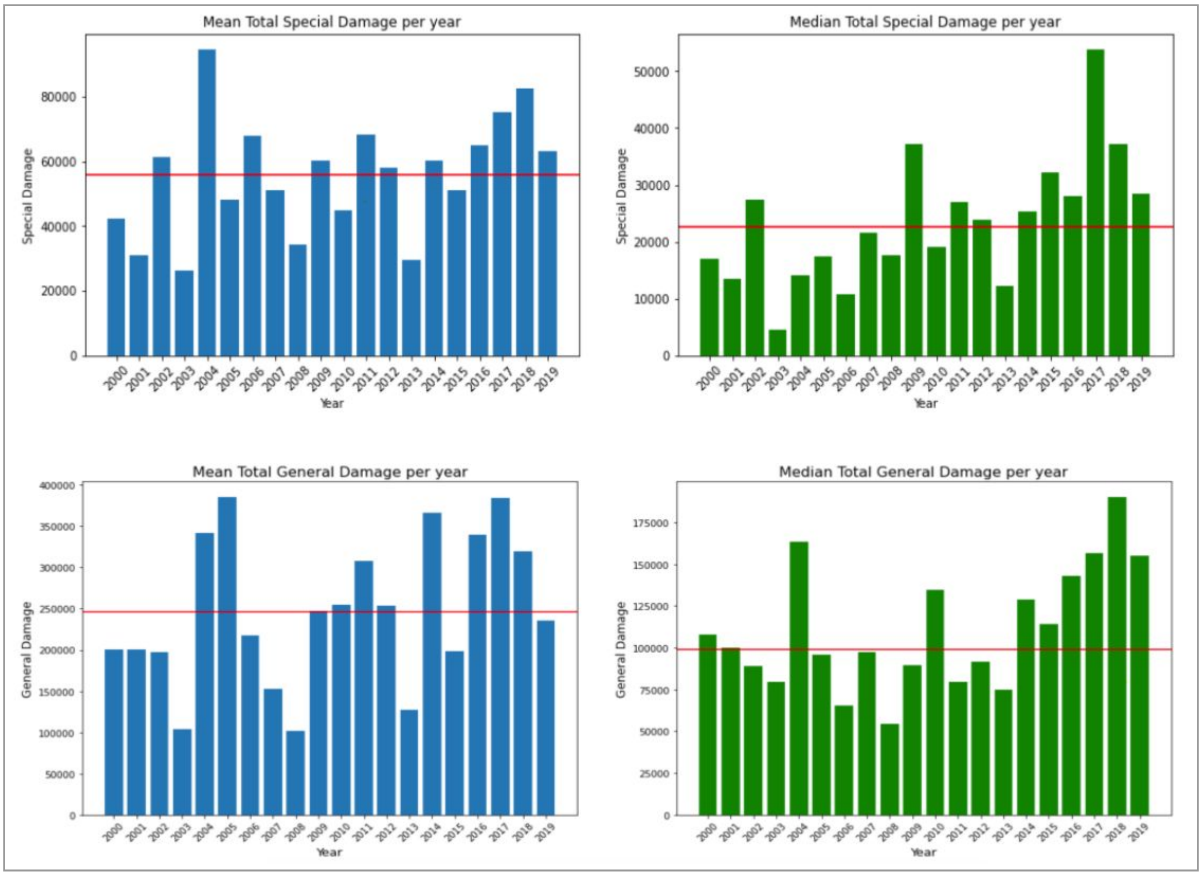


Figure 8: The mean (on the left), and Median (right) of Special damage (top) and General damages(bottom) from year 2000 to 2019. Year on x-axis and damage amount on y_axis.

The only type of damage that did not show an increasing trend in the predictions (figure 9) with the logistic regression classifier and did not have a significant p-value is future care damages awarded.

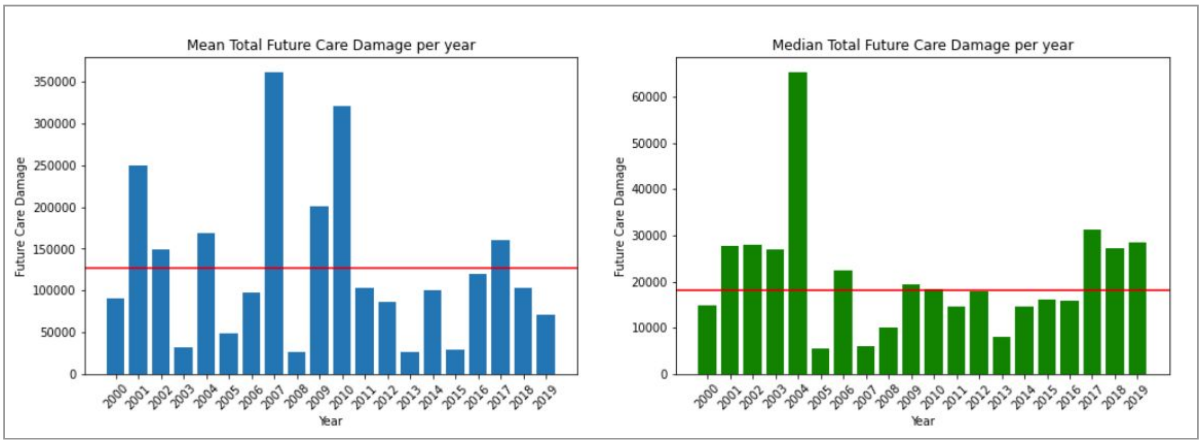
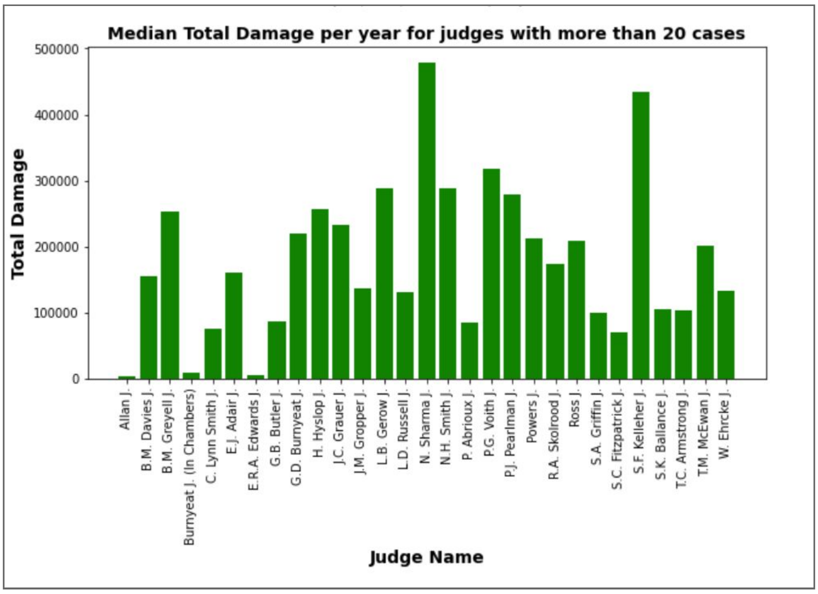


Figure 9: The mean (on the left), and Median (right) of future care damage from year 2000 to 2019. Year on x-axis and damage amount on y_axis.

The p-value for Future Care damages is more than alpha value of 0.05 therefore the results are not significant and the null hypothesis can not be rejected. The slope is not increasing year to year and the correlation of 37% might be due to cases that have a higher than average damage value therefore can not trust correlation percentage alone.

This lack of improvement over the years could be due to the fact that future care damages are under the subcategory of general damages and are mostly awarded for accidental cases and depending on the injury, the damage amount can vary a lot.

Further analysis was done on the amount of damages predicted with the classifier such as median of total damage awarded for judges with at least 20 cases in the dataset (Graph 3)



Graph 3: median of total damage awarded for judges with at least 20 cases in the dataset.

An analysis was done for the mean and median of total damage awarded for registries with more than 30 cases is shown in figure 10.

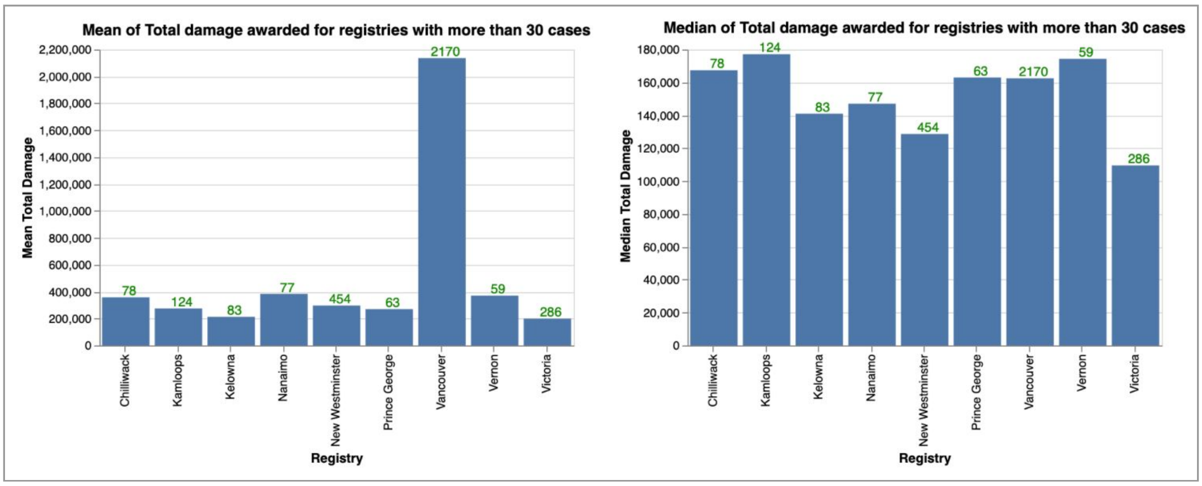
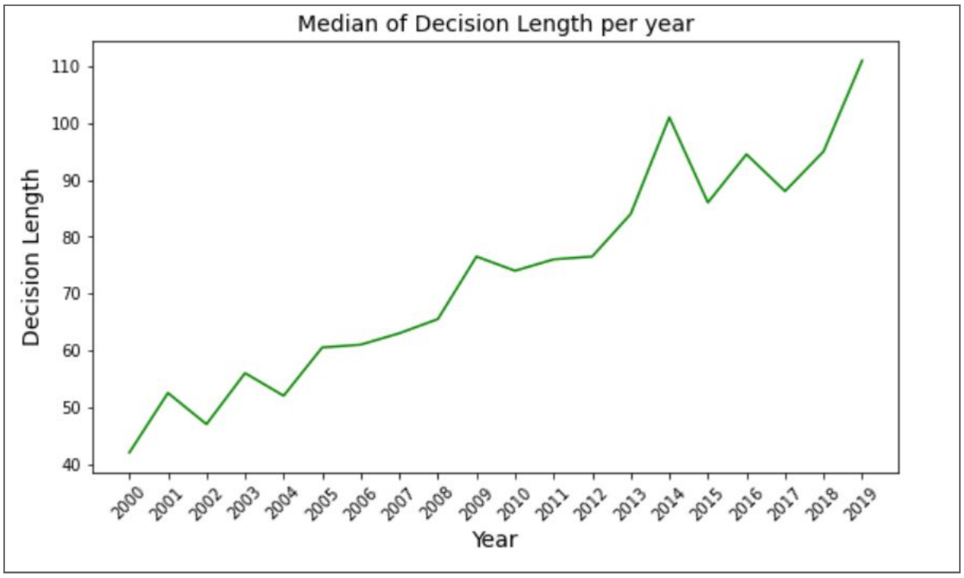


Figure 10 : Mean(left) and median(right) of total damage awarded in registries with more than 30 cases.Number on top of the bars are count of cases in each registry.

As illustrated in figure 10, Vancouver has the highest mean of total damage awarded among registries with more than 30 cases and Kamploops has the highest median of total damage awarded. This change of behaviour between mean and median is due to the fact that median eliminates the effect of outliers.

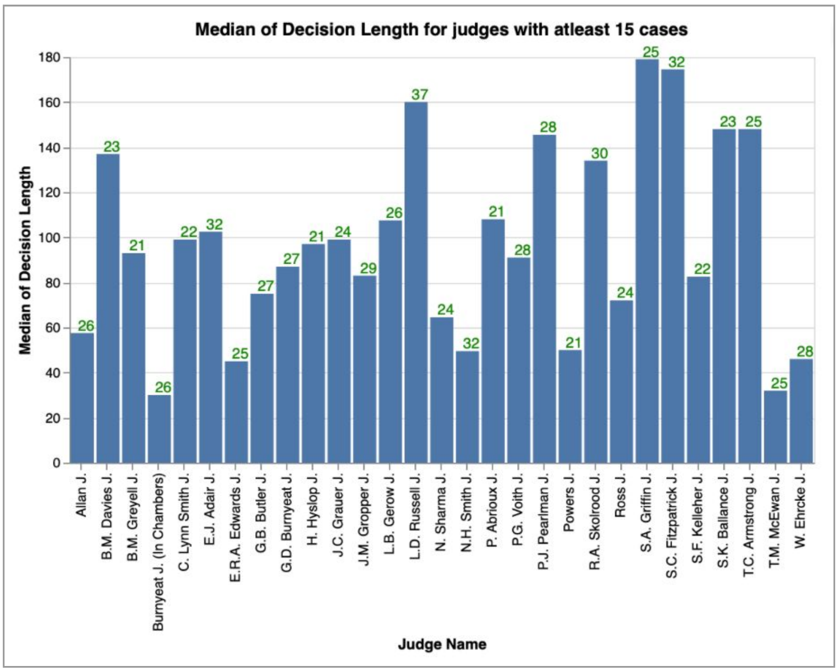
Each case is labeled with a number of paragraphs which represent the decision length. Decision lengths over the years were analysed and it showed and increased slope.(Graph 4)



Graph 4: Median of decision length over the years. Median decision length on y-axis and year on x-axis

This means that the cases have become more lengthy over the years and the it seems to be increasing more in the near future as well.

We also looked at the median of decision lengths for judges with 15 cases and more.(graph 5)
From graph 5 it is evident that judge Ross J. has the highest median of decision length followed very closely by judge Fitzpatrick J.



Graph 5: mean decision length for judges with 20 cases or more. Bars are annotated with the number of cases for each judge in green number.

In this project we also analysed and predicted contributory negligence to answer the research question “What percentage of cases have contributory negligence and what percentage were successful?”. Two main approaches of rule base approach and classification approach were used to analyse contributory negligence . (see method section)
The logistic regression classification method showed better results compared to the rule based method with 87.8% filled accuracy.

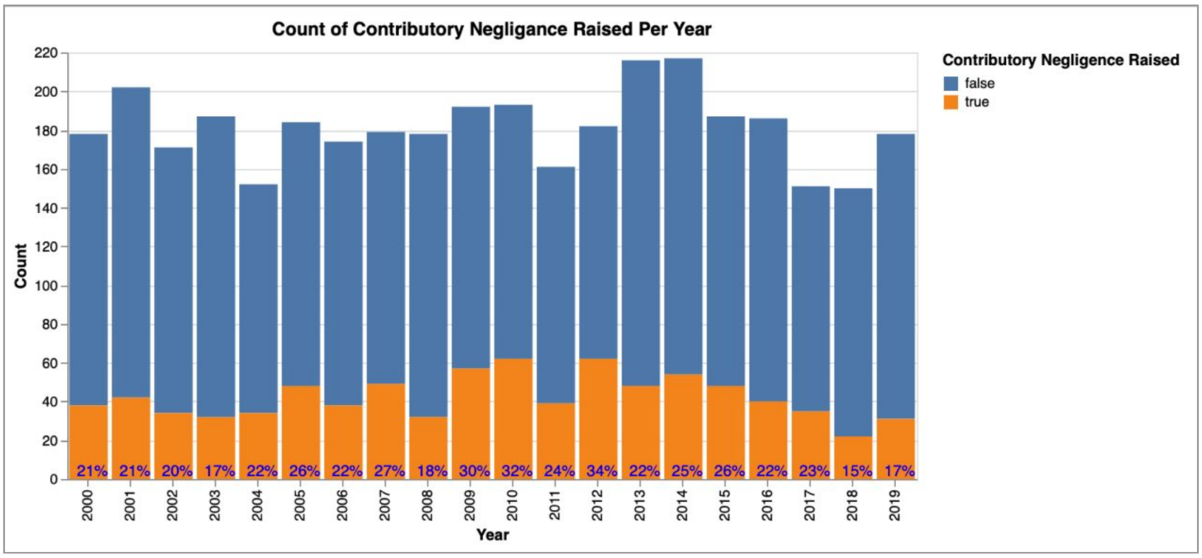


Figure 11: Count of contributory negligence raised per year (Orange) layed on top of count of contributory negligence not raised (blue) per year. The percentage of contributory negligence raised each year is marked in blue percentages on top of graphs.

Figure 11 illustrates the portion of cases where contributory negligence was raised, meaning that the plaintiff was susceptible to being charged with contributory negligence. On average 23% of cases have contributory negligence raised in them.

Next we looked at what percentage of the cases where contributory negligence was raised in were successfully charged with contributory negligence. Figure 12.

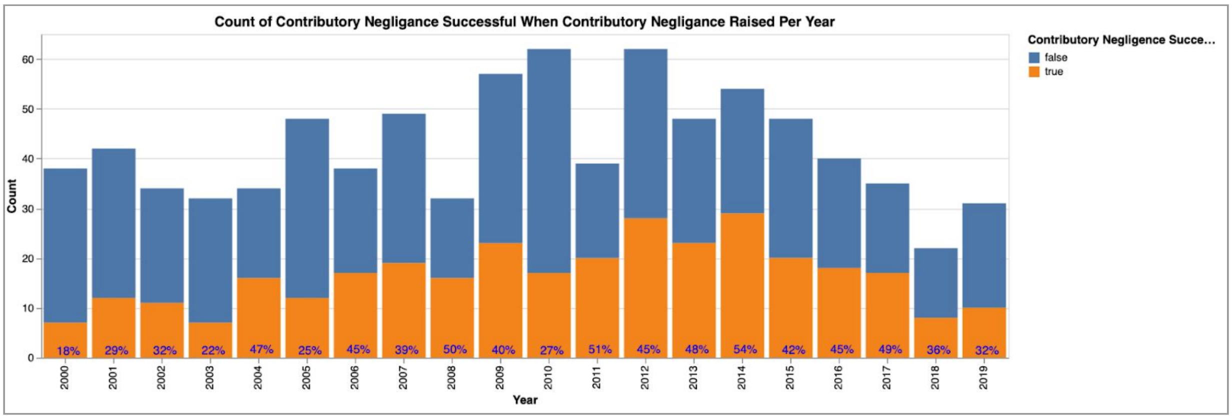


Figure 12: Count of cases where contributory negligence was successful when it was raised (orange) and contributory negligence was unsuccessful when it was raised (blue). Percentages indicate percentage of cases where contributory negligence was successful when it was raised.

Figure 12 displays a count of cases being charged with contributory negligence, meaning the plaintiff was at fault for some or all portions of damage, per year when contributory negligence was raised. On average 34% of cases when contributory negligence was raised where successful with charging the plaintiff with contributory negligence.
The trend of percentage of contributory negligence cases seem to be steady over time and do not show an increase or decrease of percentage assigned to plaintiffs.

In this project the median of total damages awarded each year when contributory negligence was successful and not successful was analysed and it is shown in Figure 13.
The mean of total damages awarded per year when contributory negligence is unsuccessful is higher than when contributory negligence is successful which can be due to the reduction of damage amounts because of contributory negligence.

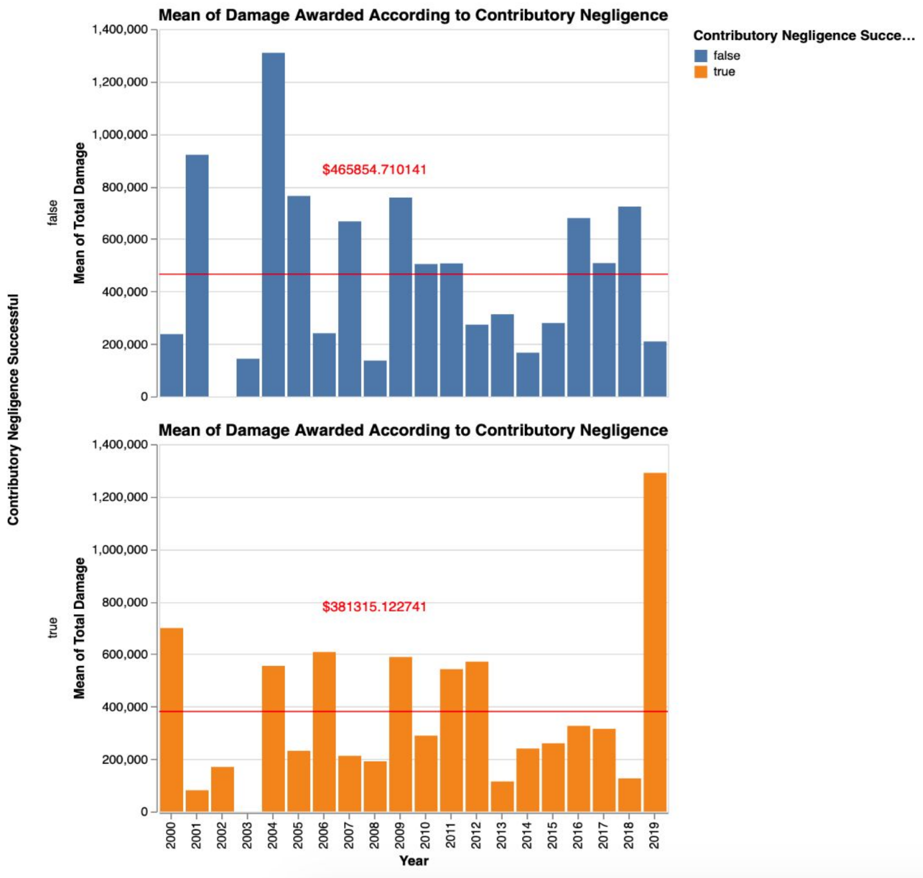
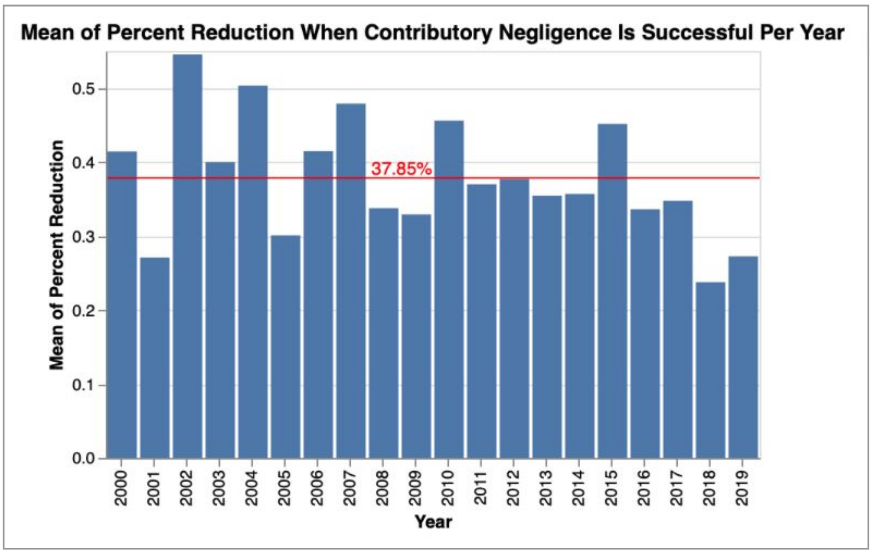


Figure 13: Illustrates the mean of total damage awarded per year when contributory negligence was successful and unsuccessful. Red lines show the mean of total damage awarded over the years.

The mean of percentage reduction due to contributory negligence was analysed and graph 6 shows 37% mean of percentage reduction over the years. This means on average when the plaintiff was charged with contributory negligence, 37% reduction was applied to the total damage amount.



Graph 6: Mean of percentage reduction due to contributory negligence from year 2000 to 2019.

To conclude, Total damage awarded to plaintiffs has been increasing from 2000 to 2019. Sub category damages of non pecuniary, total pecuniary, general and special damages are increasing over time. The only sub category that did not show an improvement over time is future care, that may be due to the fact that future care is a subcategory for general damages and it's only awarded in specific situations. On average 23% of cases have contributory negligence raised in them of which 34% were successful in contributory negligence. The trend of contributory negligence is steady and does not show and increase or decrease over the years. On Average the 37% reduction was applied to cases where contributory negligence was successful.

Future Work

Due to the time constraints related to the project we were unable to try some approaches. This section is dedicated to listing other potential approaches to explore that may result in better model performance. Increased confidence in the model would result in an increased confidence in the analysis stage.

Data

In the data section of this paper, the rules for filtering out types of cases that would not award any damages were listed. There may be other types of cases that fall into this category and could be removed from the overall dataset given enough time and resources. The amount of these cases will be low considering most of them have already been identified and filtered. Due to time constraints there was no justification to continue searching for these case types due to the relatively low impact it would have on the analysis.

LexisNexis is our only source of case data for this paper. Another legal research resource where cases may be retrieved from is WestLaw. Given more time, our plan was to include cases from both resources which could help improve our model results and help with overall analysis. There may be some overlap between the two resources but we are simply looking to increase the number of unique negligence cases that are available digitally.

Another improvement that can be made that is related to data is the order in which cases were annotated. Cases were selected manually, rather than randomly, to be annotated. Completing annotations took up approximately three to five days for each group member. This is due to the annotator needing to fully understand each case that they are annotating which can take extended periods of time depending on the case length and complexity. Since cases were selected from LexisNexis using a query there may be some ordering involved which was not taken into consideration. In the future, it would be preferable to randomize the assignment of cases to each annotator to get a more unbiased set of annotations. Furthermore, it would be preferable to annotate more than the 298 cases that were done in this paper. There were a total of 3,673 cases available to us which means that we had only 8.1% of the data annotated to train and test our machine learning model. It would have been preferable to increase this percentage to be between 15% and 20%.

Classifier

The methods section explains the use of a classifier based approach to perform information extraction. After optimizing our best classifier and achieving the best results that we possibly could, the next step would be to try using an artificial neural network as our classifier. The idea behind an artificial neural network is to encode an input into a matrix of values and pass the values through a "network" which will apply different mathematical operations at each layer of the network. Each layer will update how it interacts with the input values as it is being trained with more and more data which will eventually result in a more reliable network.

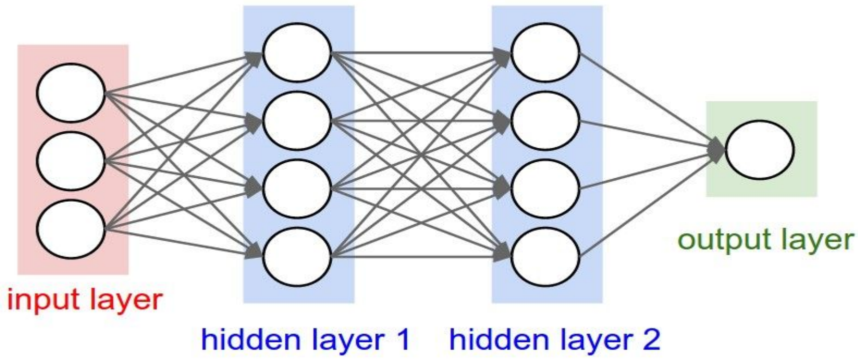


Figure 14 - Example Neural Network (image source)

Artificial neural networks perform best with large sets of data which would have required an increased amount of annotations. Resources such as Amazon Mechanical Turk provide a way to pay for the crowd sourcing of data annotation if time is more important than cost. The overall approach would remain the same as our current method, the goal would be to feed the network a sentence or paragraph into the input layer containing a monetary value or percentage we are trying to identify, which would be predicted in the output layer.

Another problem that is faced, other than time constraints, is that neural networks run fast on graphical processing units (GPUs) and relatively slowly on a central processing unit (CPU) due to the large number of matrix operations it needs to perform. All computers are equipped with a CPU and only some computers are equipped with a GPU; the faster a GPU is the better it is at training neural networks. We did not have access to any powerful GPU during the course of the project and would have had to rely on free solutions such as Google Colab which is not ideal due to resource constraints imposed by Google. Therefore, without access to a GPU the time it would take to do tuning and testing of the model would be too much to be feasible for this project.

Timeline

At the beginning of the project our team laid down a 5 week schedule for the tasks that require completion in order to deliver a successful outcome to the project. For the most part we were able to follow our weekly schedule but we were ahead of schedule for some tasks and behind schedule for others. Below we have included our updated timeline which highlights the main tasks completed by each group member on a weekly basis. Most experimental tasks or time spent researching was not included in the table for readability purposes.

	Ravi	Ilana	Niki
Week 1 **(May 4 - 8)**	Completed expected deliverables section of project plan. Completed conversion of .DOCX to .TXT Began work on determining if there are multiple defendants in a case (rule based) Extracted some static/non-moving information (judge, registry location)	Completed methods section of project plan. Began work on extracting extracting contributory negligence percentages (rule based) Extracted some static/non-moving information (case title, year)	Completed description of problem, dataset, and schedule sections of project plan. Began work on extracting if plaintiff wins and whether the case has a written decision (rule based)
Week 2 **(May 11 - 15)**	Completed extracting multiple defendants (rule based) Completed first pass of damage extraction (rule based) Created evaluation function for future use	Completed first pass of contributory negligence extraction (rule based)	Completed extracting plaintiff wins (rule based) Added 100 additional annotations for our final .CSV format
Week 3 **(May 18 - 22)**	Annotated damage tags in ~43 cases First Pass Feature Engineering Tree based classification for damages (XGBoost)	Annotated damage tags in ~43 cases First Pass Feature Engineering Linear classification for damages (Logistic Regression & SVC)	Annotated damage tags in ~43 cases First Pass Feature Engineering Tree based classification for damages (Decision Tree & Random Forest)
Week 4 **(May 25 - 29)**	Updates to evaluation methods to include more fine-grained info Convert our data into .CSV that matches external partners format First pass classification for contributory negligence	Bag of words feature engineering Further work with tuning damage classifier with Logistic Regression (best model)	Annotated percentage tags in 129 cases Begin work on generating visualizations based on results so far
Week 5 **(June 1 - 5)**	Filtering & excluding newly discovered irrelevant cases	Fine tuning contributory negligence classifier using Logistic Regression Moving code from Jupyter Notebook into .PY file	Further work on generating visualizations Created presentation for Lachlan

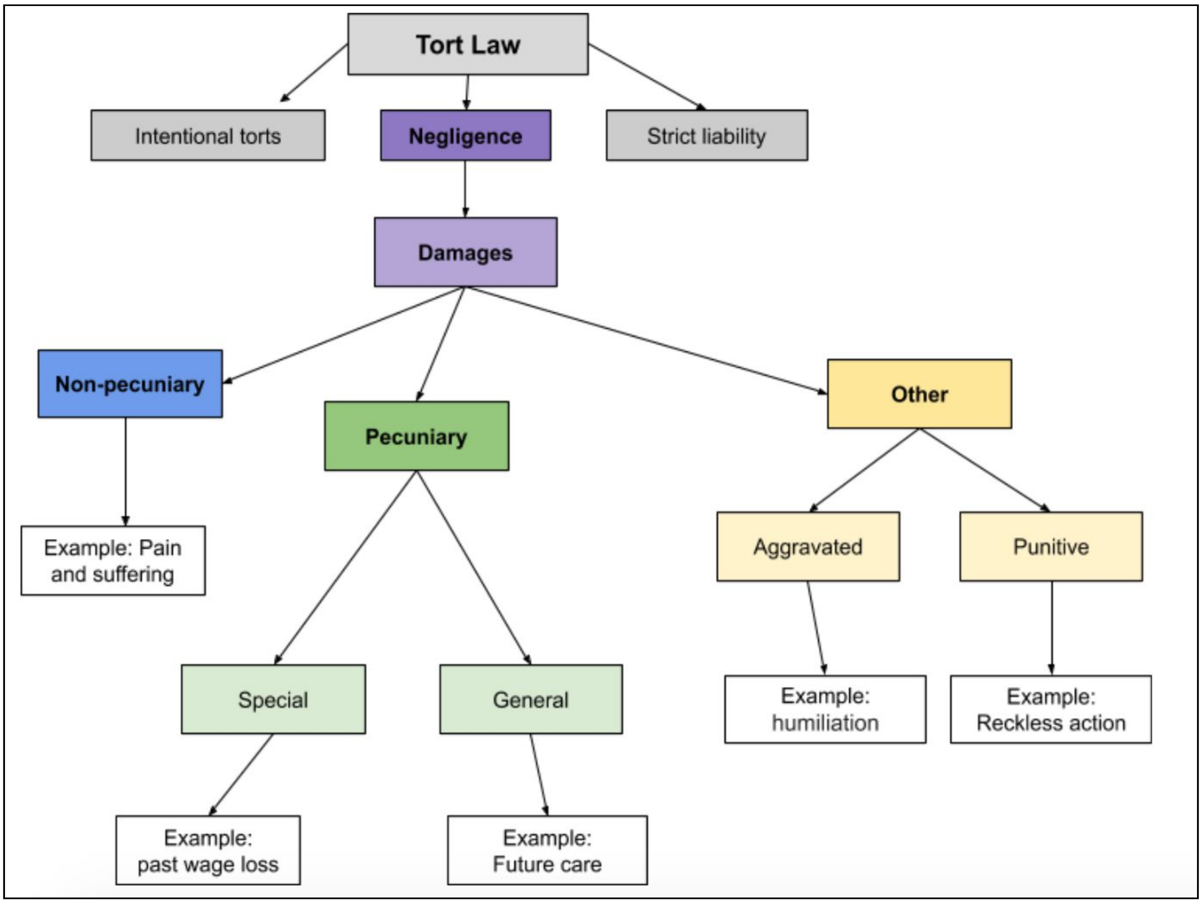
Week 6 **(June 8 - 12)**	Begin on summary, data, and timeline sections of final report Create some visualizations	Begin on methods section of final report Proving statistical significance & plotting distributions	Begin on analysis & evaluation & background section of final report More work on generating visualizations for final report
Week 7 **(June 15 - 19)**	Added final features for slight improvements to classifier Writing future work section of report Some proof-reading & minor edits of report	Ran ablation study on features	Final report review Creating presentation slides

Table 10 - Timeline of Events

Appendices:

Appendix 1:

The flow chart illustrating the sub categories of tort law and damage types of negligence category.



Flow chart of branches of BC law and the damages for tort law division.

Appendix 2:

Terminology

Bag-of-words (BOW): A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things: A vocabulary of known words. A measure of the presence of known words.

Contributory Negligence*:* "Failure of an injured plaintiff to act prudently, considered to be a contributory factor in the injury suffered, and sometimes reducing the amount recovered from the defendant."

Defendants: Defendant is a person accused of committing a crime in criminal prosecution or a person against whom some type of civil relief is being sought in a civil case.

P-value*:* "The probability of obtaining test results at least as extreme as the results actually observed, assuming that the null hypothesis is correct."

plaintiff : A plaintiff is the party who initiates a lawsuit .

Regular expressions: A regular expression is a sequence of characters that define a search pattern.

Statistical Inference: "The theory, methods, and practice of forming judgments about the parameters of a population and the reliability of statistical relationships, typically on the basis of random sampling."

Train set: is the material through which the computer learns how to process information

Development set[15] : "A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's hyperparameters."

Test set: data set that is used to test a machine learning program after it has been trained on an initial training data set

[15] ***Development set***

Appendix 3:

Case Format Example

Mclaren v. Rice, [2009] B.C.J. No. 2108
British Columbia Judgments
British Columbia Supreme Court
Vancouver, British Columbia
T.R. Brooke J.
Heard: November 24-28 and December 11, 2008.
Judgment: October 26, 2009.
Docket: M063770
Registry: Vancouver
[2009] B.C.J. No. 2108 | | |
Between Matthew Robert Joseph Mclaren, Plaintiff, and Jacob John Rice, Michael John Rice and Hilltop Gardens Farm Ltd., Defendants
(46 paras.)
Case Summary
...
...
HELD: The defendants were jointly and severally liable.
There was no evidence of any brake marks on the road...
...
...
Reasons for Judgment
T.R. BROOKE J.
1 This action arises out of single-vehicle accident, which occurred on the morning of February 26, 2005, when a three-quarter ton Ford truck owned by the defendants, Michael John Rice and Hilltop Gardens Farm Ltd., and driven by the defendant, Jacob John Rice, went off the left side of the highway, struck a wooden utility pole and rolled at least two times before coming to rest on its wheels. The issues of liability and quantum were severed and the only issue before the court is that of the defendants' liability, if any, and the contributory liability of the plaintiff.
2 ...
...
...
45 I am satisfied that the defendants have failed to establish on a balance of probabilities that the plaintiff was not wearing a seatbelt or that the seatbelt available to him was in working order, or if it had been in working order, that it would have prevented or reduced his injuries.
46 In the result, I find the defendants jointly and severally liable to the plaintiff for such loss and damages he sustained as a result of the accident. Given this result, I do not consider that it is necessary for me to remain seized of the assessment of damages. The plaintiff will have his costs at Scale B.
T.R. BROOKE J.
End of Document

Appendix 4:

An example of case annotation at high level:

Case Name	Graham v. Lee, [2004] B.C.J. No. 2052
Written Decision?	Y
plaintiff Wins?	Y
Multiple defendants?	N
Judge Name	C. Lynn Smith J
Decision Length: paragraphs)	56
Registry	Vancouver
$ Damages total before contributory negligence	215000
$ Non-Pecuniary Damages	90000
$ Pecuniary Damages Total	125000
$ Special damages Pecuniary (ie. any expenses already incurred)	0
Future Care Costs (General Damages)	0.00
$ General Damages	125000
$ Punitive Damages	0
$Aggravated Damages	0
Contributory Negligence Raised?	Y
Contributory Negligence Successful?	Y
% Reduction as a result of contributory negligence	50
$ Reduction as a result of contributory negligence	43000
$ Final Award after contributory negligence	172000