Outcome-Based Judgement Categorization of the Supreme Court of Canada

by

Thomas Malley

A thesis submitted in partial fulfilment

of the requirements for the degree of

Master of Science (MSc) in Computational Sciences

The Faculty of Graduate Studies

Laurentian University

Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Office of Graduate Studies/Bureau des études supérieures

Title of Thesis Titre de la thèse	Outcome-Based Judgement Categorization of the Supreme Court of Canada	
Name of Candidate Nom du candidat	Malley, Thomas	
Degree Diplôme	Master of Science	
Department/Program Département/Programme	Computational Sciences	Date of Defence Date de la soutenance September 12, 2022

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Ratvinder Grewal
(Supervisor/Directeur(trice) de thèse)

Dr. Bernadette Schell
(Committee member/Membre du comité)

Dr. Kalpdrum Passi
(Committee member/Membre du comité)

Approved for the Office of Graduate Studies
Approuvé pour le Bureau des études supérieures
Tammy Eger, PhD
Vice-President Research (Office of Graduate Studies)
Vice-rectrice à la recherche (Bureau des études supérieures)
Laurentian University / Université Laurentienne

Dr. Aniket Mahanti
(External Examiner/Examinateur externe)

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

Outcome-based judgement categorization of the Supreme Court of Canada (SCC) focuses on the multidisciplinary field of computational law. Regarding court hierarchy, the SCC is the highest court in Canada. Decisions from this court generally bind any lower court. Since court decisions are in a textual format, it is possible to correctly categorize outcomes of the SCC utilizing Natural Language Processing (NLP) techniques. The experiment contained shows algorithmic categorization performance F1 greater than 60. This result is significant given the binary nature of case outcomes (allow, dismiss) that an individual unfamiliar with the law should be able to guess 50% of the time correctly. This work is a preliminary study of future work to indicate the possibility of outcome forecasting in the judicial branch of the government.

Keywords

Natural Language Processing, Canadian Law, Court Case Categorization, Supreme Court of Canada, Precedence.

Acknowledgements

First, I take this opportunity to express my deepest gratitude to my supervisor Dr. Ratvinder Grewal, for his continuous support and for allowing me to explore this field. Freedom of academic exploration with guidance and direction has allowed me to grow as an academic and accumulate new knowledge and skills.

I cannot begin to express my thanks to Rachelle Barrette for all her love and support.

I am also grateful to the CHILL research group for participating in discussions, asking questions, and help over the past few years.

Finally, I would like to extend my thanks to all those who were directly or indirectly involved. Including but not limited to my friends and family for their continuous support, as well as all of my professors who have guided me during my studies.

"Persistence without insight will lead to the same outcome."

- The Armourer (The Book of Boba Fett S1.Ep5: Return of the Mandalorian)

Contents

Glossary

CanLII	A non-profit organization hosting a database containing judgements from all Canadian Courts..
Corpus	is a collection of texts. It includes all written works in a by an author, in a domain, or body of knwoledge..
Open Data	is described as data that is freely accessible, reusable, reproducable. Open Data should also be in a computer readable format for bulk access..
Precedent	is known as a source of law in Canada. Previsouly decided cases hold authority over preceeding case. New decisions should follow previous decisions..
Recidivism	the likelyhood that someone convicted of a crime will reoffend..

Tribunal	persons or groups authorized to make decisions or judgements within a specific domain..

Acronyms

AI	Artificial Intelligence.
BNAA	British North America Act.
BOW	Bag of Words.
CJC	Canadian Judicial Council.
KNN	K-Nearest Neighbour.
LR	Logistic Regression.
ML	Machine Learning.
NB	Naïve Bayes.
NLP	Natural Language Processing.
POS	Parts of Speech.
SCC	Supreme Court of Canada.

SVM	Support Vector Machine.
TF-IDF	Term Frequency - Inverse Document Frequency.

List of Tables

List of Figures

Chapter 1

Introduction

The multidisciplinary field of computational law is the area of interest presented in this thesis. In the past, law, legal research, and legal reasoning, for the most part, were manual endeavours. Lawyers, paralegals, and other legal researchers must manually comb through Canada's two primary sources of law: statutes and case law. Significant technological strides have made it not uncommon for Artificial Intelligence (AI) and NLP techniques to be leveraged in law and legal research. There are currently self-driving cars, automated personal assistants, smart home devices, the automation and simplification of legal research, and other applications that have been common or relatively common for several years. There are also domain-specific applications of AI decision-makers, some known as Expert Systems.
This work contributes to the body of authors such as [36, 38, 3, 52, 59, 46, 35, 25, 32, 37, 48] by replicating similar techniques with new untested data from the SCC. This work also proves the feasibility of utilizing SCC case data as a training set for NLP as well as lays the groundwork for future researchers to use SCC data for machine learning. The rest of the thesis is structured as

follows. chapter 2 discusses relevant background research. chapter 3 discusses the methodology of the implementation. chapter 4 discusses the results of the implementation. Finally, chapter 5 and chapter 6 discuss the results and future work.

1.1 Canadian Law

What makes the judiciary particularly interesting to study, especially in Canada, is that there are few members of this exclusive club.[10] This means we can take a close look into a judge's background and attempt to determine any socializations that may have occurred.[10] Many people do not understand the judiciary; society does, however, accept that a judge's decision in the courtroom is final. Those in the sphere of law who study the judiciary have come to understand that a judge is still human. It is known that a judge will carry his/her beliefs to the bench they may have had prior to their appointment.[10] Knowing the history of the judiciary would allow us to understand how modern benches may perform in the future.[10] At face value, if we were to digitize the judiciary, we would hopefully be able to avoid personal belief altogether. Bias and the judiciary are discussed further in section 2.2.

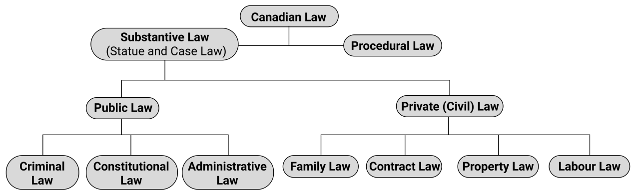
1.2 Expert Systems

In the 1950s, British mathematician Alan Turing proposed what is known as the “Turing Test”. This is the first time in history that we see the theory behind machine intelligence and the beginning of the artificial intelligence revolution. The Turing Test assesses whether a machine can think and whether a machine could be indistinguishable from a human in conversation.

This game is simple; it requires a machine (A), a human(B), and a human interrogator(C). The game's objective is for participants A and B to fool C into believing they are human; the interrogator C must question A and B and determine which is human and which is the machine.[45] This "Can machine's think" theory is becoming more of a reality fifty years later as our technology progresses. We currently employ this concept in Machine Learning (ML) and AI; specific use cases are known as expert systems. An expert system, in simple terms, is a decision-maker that applies previously gained knowledge to make a decision. An expert system can be compared to, in some sense, a judge. The expert system and judge are decision-makers; their purpose is to make decisions based on information and knowledge. It is interesting to theorize the possibility of digitizing the government's judicial branch by creating an expert system judge. If there was in existence an expert system that was able to pass a Turing Test (meaning it was indistinguishable from a human), would it be suitable as a replacement for a judge? A judge must differentiate between law and fact; they must also be knowledgeable in the law and its application. A theoretical expert system judge should be able to complete the same functions. "If a system reliably yields opinions we view as sound, we should accept it..." [60]. The complexity of the law is not found in other domains. Figure 1.1; public law is different from private law. Criminal law has degrees; it is not enough to consider if the law is broken, but how much it was. Other areas of law, such as tax law, are relatively absolute.

For an expert system to be built, there are a few requirements, such as knowledge acquisition, representation and encoding. These processes are described further in section 2.3. Following this, we have testing, evaluation, and finally, implementation. It is easy to theorize about these concepts as they

Figure 1.1: Divisions of law in Canada



seem relatively straightforward; in reality, however, a system like this would require years of development and the efforts of multidisciplinary experts.

1.3 Objectives

This study investigates popular NLP techniques and their effectiveness in classifying court cases from the SCC. There are two main objectives of this study, as listed below.

1. Investigate the feasibility of utilizing the cases from the SCC as a dataset for NLP.

2. Determine an optimal machine learning algorithm and parameters (See section 2.3).

These objectives will help inform the criteria for a more extensive study. First, it is essential to understand the feasibility of utilizing the SCC cases as a dataset and how varying machine learning algorithms and settings perform on the dataset. Future implementations of this work would extend beyond simple classification techniques and move towards outcome forecasting. Outcome forecasting is discussed further in section 2.1 and chapter 6. This work should also help foster discussion toward the accessibility of court cases in a computable format. Finally, it should be noted that there were limitations to the experiment that are discussed further in section 3.3

1.4 Research Questions

This study aims to ascertain the viability of using the SCC Court Cases as a dataset for court case outcome forecasting (more details in chapter 6).
In order to understand if outcome forecasting is possible with the SCC data available, it must first be understood if we can perform the more straightforward task of outcome categorization (section 2.1). Court outcome categorization should inform if there are patterns in the data that we can leverage. If we took a subset of SCC cases and had participants randomly guess if the outcome of the case appeal were dismissed or approved, we would expect a 50-50 correct vs incorrect distribution. For this reason, algorithmic performance above 50 would indicate enough patterns in the data to leverage in future projects (See chapter 6).

R.Q1 Can we classify outcomes of the Supreme Court of Canada with some degree of accuracy above a 50-50 guess (above 50% F1 Score)?

A secondary interest in this study relates to the chosen algorithm. section 2.1 literature review outlines multiple algorithms tested in the experiment outlined in section 3.4. It is typical for the SVM algorithm to have the maximum performance [36, 38, 56, 59, 46, 35, 32, 37]. For this reason, a second research question is posed to identify the best-performing algorithm.

R.Q2 Which algorithm obtains the best performance for this dataset?

NLP and machine learning algorithm parameters are adjusted and tuned to optimize performance. Therefore, understanding which parameters contribute the most to algorithmic performance is of interest.

R.Q3 Which algorithm parameters contribute the greatest to algorithmic performance.

The three research questions, as mentioned earlier, will help frame future work as described in chapter 6.

Chapter 2

Literature Review

The Canadian government as a constitutional monarchy includes multiple levels of government: Federal government, Provincial, territorial governments, and municipal or regional governments. Each level of government has a legislative branch responsible for representing the citizens and holding the executive branch accountable. The executive branch is responsible for creating and administering policies, whereas the Judicial branch of the government is responsible for developing law and resolving disputes. During the execution of duties, the Judicial branch of the government creates what is known as common law. All law, including common law, is textual. Given the word-based nature of law, it would be possible to apply NLP techniques to the law. section 2.2 further describes the Canadian court system and the role of Common law and precedent in Canadian law. section 2.2—2.2 further describes the features of Canadian legal texts, and finally, section 2.2—2.1 describes how computer science can leverage legal texts and how we can potentially move towards outcome forecasting.

2.1 Previous Works

Discussions around utilizing computers to predict court cases have been discussed since the early 1960s as mentioned in [3]. Indexed legal databases such as Westlaw and LexisNexis have existed in one form or another since the '90s [36]. More recently, there has been an uptick in practical research on automated court case prediction as technology and AI has improved. Previous works by [36, 38, 3, 52, 59, 46, 35, 25, 32, 37, 48] in this discipline all follow a similar implementation pipeline. Table 2.1 summarizes similar works by other authors, the data set they used, which algorithms were presented, and how the performance was reported. Some authors reported a final result [3, 27, 36, 37, 38, 36, 37, 38, 56], where other authors [32, 52, 59] did not report a final score but rather described a series of results. In short, a dataset is chosen from either a Supreme Court[56, 59] or a tribunal such as the European Court of Human Rights [3, 36, 38, 37]. The raw text is then extracted and processed using one or more techniques outlined in section 2.3. The processed data is then transformed into the vector space model using either the Bag of Words (BOW) or Term Frequency - Inverse Document Frequency (TF-IDF) methods. The transformed data is used to train various ML algorithms such as Support Vector Machine (SVM), Naïve Bayes (NB), Logistic Regression (LR), and K-Nearest Neighbour (KNN). Algorithmic performance is typically measured by an F1 Score, as described in section 2.3. It was not uncommon for other authors to achieve an F1 performance above a 50%. 50% is a significant cut-off point since the implementation is binary "[appealed, dismissed], [violation,non-violation], .ect ".

Reference	Dataset	Algorithms	Reported Performance
[3]	ECtHR	SVM	79 (F1)
[27]	US Supreme Court	SVM	70 (Accuracy)
[32]	Summary Paper	K-NN, LR, RF, SVM	Multiple (Accuracy)
[36]	ECtHR	SVM	75 (Accuracy)
[37]	ECtHR	SVM, H-BERT, LEGAL-BERT	66 (F1)
[38]	ECtHR	SVM	65 (F1)
[52]	BGH (German Court of Justice)	SVM	Multiple (F1)
[56]	French Supreme Court	SVM	96 (F1)
[59]	Philippine Supreme Court	SVM	Multiple (Accuracy)

Table 2.1: Summary of Related Works

Prediction vs Classification

Typically, authors in this research domain tend to use different terms interchangeably. [39] argues that most authors take one of 3 approaches:

1. Outcome identification: meaning to identify the verdict and can be done via a keyword search. No AI is necessary for this method.

2. Outcome-Based Judgement Categorization: This method utilizes textual information to categorize a judgement.

3. Outcome Forecasting: The most interesting of the three approaches predicts future decisions of a court before a decision is made. In the case of the SCC, we would use all filings and court documents before the SCC hears the case. This would exclude the discussion of the outcome of judicial opinions that may pollute the performance of an algorithm.

From reviewing authors in this field, many authors claim they are forecasting. However, they are simply performing Outcome-based Judgement Categorization. Figure 2.1 [39] illustrates the three distinct pipelines. chapter 3 focuses on outcome-based judgement categorization, whereas chapter 6 discusses forecasting future decisions.

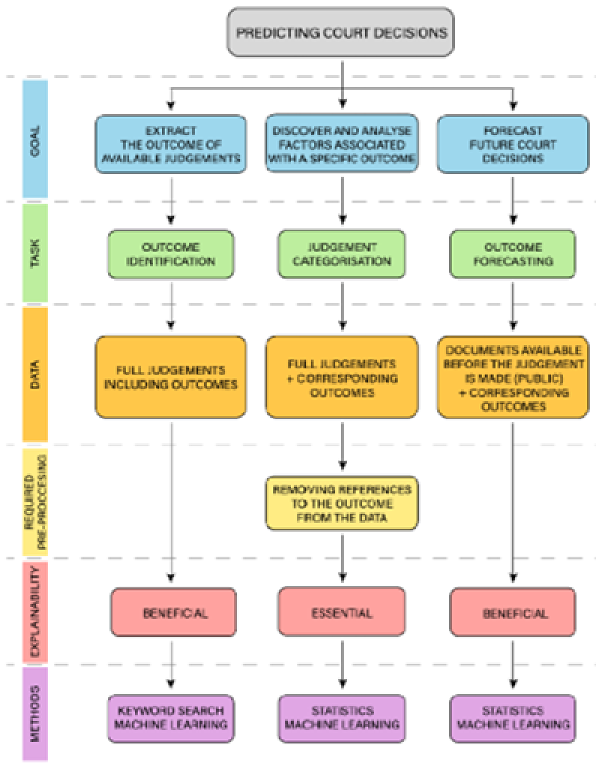


Figure 2.1: Three Methods of Predicting Court Decisions [39]

2.2 Background Material

Canadian Court System

The Judicial branch of the Canadian government as the court system can be simplified into four hierarchical categories. The lowest court in the land is the provincial courts. Provincial courts can be divided into separate courts; however, this differentiation is out of scope. After provincial courts, there are superior courts which have increased responsibility in decision making. Lower provincial courts can generally appeal to superior courts.[20] Again, the differentiation of courts is beyond the hierarchy is not in scope.

After the Superior courts, there is the Court of Appeal. Courts of appeal are the highest Provincial courts and will hear appeals from the provincial and superior courts. The SCC is the highest court in Canada and only hears appeals from lower courts. It is made up of 9 judges. They receive, on average, 500-600 applications per year but only hear 65-80 of them. Unlike other courts, the supreme court chooses what cases they hear. All cases are appeals, meaning there is already a decision in a lower court that the applicant is looking to have overturned. Since the Supreme Court is selective on the cases they hear, it is only cases of high importance or those where the outcome will significantly affect society. [21] The importance of court hierarchy (section 2.2) in this context involves the role of precedent. Typically Judges in lower courts are bound by the decisions of upper courts. The SCC, the highest court in the land, sets the standard to which the lower courts must adhere. Precedent is the main reason the SCC court cases were selected as part of the experiment in chapter 3.

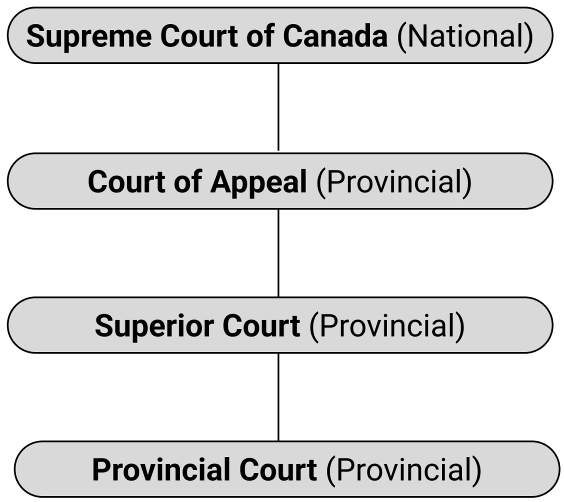


Figure 2.2: Simplified Hierarchy of the Canadian Court System

Judicial Independence

Judicial Independence means that the Judicial branch of the government can freely make decisions without influence or interference. In some instances, a Judge may be required to make an unpopular decision and is therefore protected. The Constitution Act of 1867 (British North America Act (BNAA)) defines a mandatory retirement age for Judges as 75. Other than retirement, the only way a judge can be removed is if they violate "good behaviour", which is not well defined. It should be noted that removing a Judge is difficult by design, so politics or politicians cannot sway Judges.

Law in Canada

The two primary sources of law in Canada are Statues and Case Law. Statues are laws enacted by parliament. They "prescribe conduct, define crimes, create inferior government bodies, appropriate money, and promote public good and welfare" [13]. Case Law is a little different; case law, also known as common law, is judge-made. A formal definition is "Law based on judicial precedent, rather than legislative enactments". [13] This means that if a judge decides on a case, later cases of similar nature should follow similar decisions; older decisions are used to help make current decisions (Precedent). For example, if there exists a precedent regarding animal abuse toward cats, in a case where a judge is required to decide an animal abuse case against dogs. The judge would have the discretion to follow precedent since both are four-legged furry animals kept as pets. The judge could also make a distinction and make a new decision, such as felines and canines are dissimilar and have different needs and requirements.

If similar past cases bind all future cases, then precedent strengthens the arguments of utilizing historical court cases as a training set to build a model to forecast future decisions. Forecasting future decisions is discussed further in chapter 6.

Legal Documents

As discussed in section 2.2, the law and legal language are generally in a textual format. There exist some complexities in the sentence structure of English written law. For example, legal texts have repetitiveness, Latinisms, and sometimes archaic or infrequently used terms [62]. These features of the legal text are not new and have always been present.
Some examples are:

1. Archaic or rarely used words and expressions:

- Hereinafter: "Further on in this document".
- Surrejoinder: "A plaintiff reply to the defendant's rejoinder".

- Aforesaid: “Previously mentioned”.

Archaic words and expressions are common occurrences in legal texts. These words have clear definitions for those in the sphere of law. However, the meaning is complex for those unfamiliar with the terms.

2. Foreign words and expressions (Latin).

- Attainder: “The forfeiture of land and civil rights as a consequence of a sentence of death for treason”.
- Amicus Curiae: “Friend of the Court”

Foreign words are traditionally in Latin and are common occurrences in legal texts.

3. Repetition.

There is an almost obsessive repetition of words in legal text, often to ensure that there is no ambiguity. For example, the following excerpt shown in [62] utilizes the word “chair” nine times and the word “vice-chair” four times. There is a total of 120 words in the excerpt.
“Powers of vice-chair 11. Where - (a) a member of a Board is appointed to be vice-chair either by the Assembly or under regulation 10, and (b) the chair of the Board has died or has ceased to hold office, or is unable to perform the duties of chair owing to illness, absence from England and Wales or any other cause, the vice-chair shall act as chair until a new chair is appointed or the existing chair resumes the duties of chair, as the case may be; and references to the chair in Schedule 3 shall, so long as there is no chair able to perform the duties of chair, be taken to include references to the vice-chair.” [62]

4. Complexity.

Legal text often strives for inclusiveness and to cover as many circumstances and outcomes as possible, and it is often verbose. The following excerpt illustrates lengthy and complex wording often used in legal texts. [62]

"If, after informing the supervisory authority concerned under subsection (3), any measures are taken by the supervisory authority against the insurance undertaking concerned are, in the opinion of the regulatory authority, not adequate and the undertaking continues to contravene this Act, the regulatory authority may, after informing the supervisory authority of its intention, apply to the High Court for such order as the court may seem fit, in order to prevent further infringements of this Act, including, insofar as is necessary and in accordance with the Insurance Acts 1909 to 2000, regulations made under those Acts and regulations relating to insurance made under the European Communities Act 1972, the prevention of that insurance undertaking from continuing to conclude new insurance contracts within the State4." [62]

5. Passive

Often legal texts are written in the passive tense. Some examples are "shall be used" and "may be used by". [62]

6. Impersonal writing

Law, by nature, is supposed to be impartial and authoritative. Impersonal writing reinforces these ideas, especially when the text attempts to express authority or obligations. For example, terms such as “every person”, “everyone”, “no person”, and “no one” is frequently used.

7. Nominalization

Nominalization is a writing technique in which nouns are altered to verbs, and additional words are added to convey the same meaning. For example, the verb “amend” is nominalized into “to make an amendment” [62]

Any legal text does not necessarily have each of the features presented above; however, text in the legal context may have one or more of the previously mentioned features. The complexity of legal text requires management in NLP, and the management of the text is discussed further in section 2.1.

NLP and Law

NLP is a field in which to train computer programs to understand natural human language. NLP, a subfield of linguistics and AI, allows us to create programs or applications that further human-computer interaction. Most of what we consider “The Law” is in a textual format compatible with NLP techniques. As previously defined in section 2.2, the two primary sources of law in Canada are case law and common law. There are instances where NLP and other AI techniques are used in law. For example, Intuit's TurboTax is intelligent software that aids tax preparation. Other software is designed to summarize legal documents, contract reviews, and legal research, to name a few. Given this, it is not beyond imagination to suggest utilizing the same sorts of technology to forecast future judgements of the Judicial branch of the government instead. The experiment discussed in chapter 3 —4 explores the preliminary steps required to work toward some type of expert system (section 1.1) that may be able to supplement the Judicial branch of the government.

Opportunities of NLP and Court Judgements

The first and most apparent opportunity of legalAI is that computers do not get tired. Baring potential updates and maintenance restarts, turning off a computer is generally unnecessary. With cloud computing gaining popularity, it would be possible to have cloud computing systems with 99.99% uptime. Availability of justice could be increased outside of regular operating hours beyond what society generally operates. Additionally, humans cannot be in more than one place at a time. However, a computer program can be accessed simultaneously by multiple users and from a geographic distance. Availability of the justice system by utilizing AI —based machines may help alleviate congestion in the court system. For example, it was found [30] that the average time to judgment in Ontario was around 98 days; in the federal court, the median was 163. Similarly, the Canadian government published a video regarding court delays. They noted that:

"One of the biggest problems facing Canada's Criminal Justice System is court delays. Although fewer cases go to adult criminal court, individual cases increasingly take longer to complete. Additionally, the number of people who are in jail waiting for a trial has increased, and now outnumbers the number of people who have been convicted of an offence and are serving sentences. System inefficiencies can cause people to lose confidence in the system and costs to increase." [58]

Challenges of NLP and Law

Every opportunity comes with a challenge, and utilizing AI technology in law is not without its challenges. There are multiple ethical concerns with utilizing AI in law; for example, there are concerns of bias, reverse engineering, trust, and liability. These concerns are essential topics of discussion, and solutions to the following issues are not presented and are out of scope for this work.

Bias

[7, 50, 36, 38, 40]

In the United States, there is an implementation of legal AI known as COMPAS. This system is meant to aid a judge in deciding by informing on the likelihood of Recidivism. A ProPublica study found a significant flaw in the system, “black defendants were far more likely than white defendants to be incorrectly judged at a higher risk of reoffence.”[51] It was also found that “white” defendants were more likely to be incorrectly judged as low risk.[51] It would then seem that these systems may not be reliable in law as there has been a clear history of bias in one fashion or another. The real problem with COMPAS is with the company responsible for development. The company has been “criticized for lacking transparency” since the algorithm behind COMPAS is not publicly available, meaning that there is no way for any third parties to audit the system to ensure it is free of bias.[51]

Although sometimes bias is necessary or “desired outcome”. A Gladue Report in Canada is a pre-sentencing hearing that a Canadian court can use when sentencing an offender of Aboriginal background. [64] This allows a court to consider the unique circumstances of aboriginal people when deciding.

Reverse Engineering

[40]
An algorithm is a defined set of steps —we would want transparency in a system that may make decisions in cases such as criminal law, having the ability to audit the system and ensure that it is not being manipulated in some way. However, transparent predictive systems may create an opportunity for abuse. Prosecutors who experience these algorithms over and over may devise a strategy only to prosecute cases in which they can guarantee a win given their understanding of the algorithm's inner workings, meaning they could have a 99% win rate. Conversely, sizeable criminal defence firms may use the same tactics to defend potentially violent criminals or corrupt politicians. Thereby further removing access to justice for the average person.

Trust

[50, 26, 40, 7]
In law, the decision-maker is either a judge sitting alone or with a Jury. In some cases, the decision-maker is a Tribunal. Replacing this process with a machine may not have the desired outcome. Machines cannot feel sympathy or other emotions. They are simply cold and calculating machines. Even though there is no room for emotion in the courtroom, the decision-makers still have them. Regardless, there is humanity in the decision-making process; sympathy, empathy, and facts might better allow for decisions acceptable by society over a machine decision looking at facts alone.

Liability

[42, 26, 16, 6]

Human decision-makers are also responsible for their decisions regardless of Judicial Independence; a decision that radically opposes the legally accepted framework can have consequences. Such as the case of Justice Robin Camp, who was recommended for removal from the bench by the Canadian Judicial Council (CJC). His removal from the bench followed inappropriate comments, which amounted to victim-blaming.[63] Justice Camp did not conduct himself appropriately given his authority in the courtroom and was held liable. Computer programs would require a similar authority to make decisions in law; the source of their authority and the liability for any outcomes should be questioned.

2.3 Methodology - NLP Pipeline

An NLP pipeline is a workflow that defines a set of repeatable steps for taking raw data, prepping, and transforming the data before finally training the data on a given NLP model. section 2.3 illustrates an NLP pipeline utilized in training and testing textual—based machine learning models; the following sections discuss this pipeline in further detail.

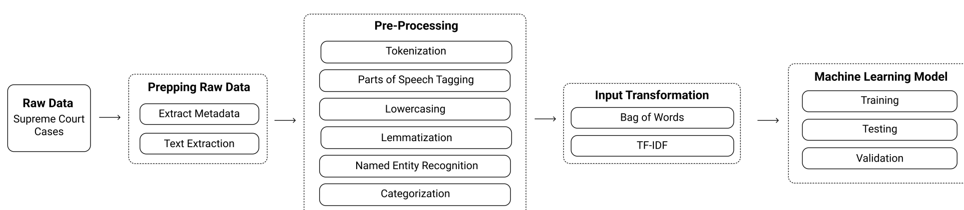


Figure 2.3: NLP Pipeline: From raw data to trained model

Data Acquisition

Data acquisition for machine learning projects can have varying levels of complexity. For AI and ML to foster, the data must be "Open and transparent" to foster research. Open Data has gained some traction in government and other organizations; however, open Judicial Data is essential for transparency, participation, and collaboration of the citizens in society [33]. For example, the SCC provides access to judgements from a repository of cases it has heard. Some websites, such as CanLII [1], allow access to judgements from all levels of the judiciary and tribunals (see section 2.2, 2.2, 2.2 for details on the judicial branch of the government.). There is no Canadian database of available judgements for either bulk download or in a computable format. Once the data has been acquired, it is necessary to prep the raw data.

Prepping Raw Data

After data acquisition, it is vital to prep the data for further processing. Data prepping is required since the data may not be in a consistent or computable format. In such cases, storing data in document format is necessary to extract the raw data, removing unimportant features such as document formatting or spacing.

Pre-Processing

The third step of the NLP Pipeline is to preprocess the raw data. There are no defined preprocessing methods, but reoccurring/common themes are seen in the literature. Pre-processing is a method in which we modify our dataset to improve model efficiency and accuracy. These methods include Lower Casing, Parts of Speech Tagging, Stop Word Removal, and Lemmatization. Before any preprocessing steps occur, tokenizing the raw text is necessary.

Figure 2.4 illustrates the tokenization process.

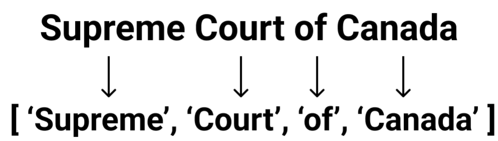


Figure 2.4: Sentence Tokenization - Splitting sentences into tokens

Lower Casing

Lowercasing is a straightforward process in which we replace all the capital letters in the text with the lowercase equivalent. For example, “Court” becomes “court”. This is a necessary step since words that begin with a capital letter would be treated as a different word compared to the lowercase equivalent.

Stop Word Removal

Stop words are the most common and reoccurring words in a language that does not provide text information. For example, in English, we have many repeating words such as “the”, and “a”. An example shown in Figure 2.5. Unfortunately, there is no widely accepted list of stop words; however, many machine learning libraries provide such lists [8, 4].

POS

Parts of speech is a process by which we identify the word type. Some word types are Nouns, Verbs, and Adjectives, to name a few. Utilizing neighbour words, Parts of Speech (POS) taggers will mark words with a designated category. For example in Figure 2.6, in the sentence, “The quick brown fox

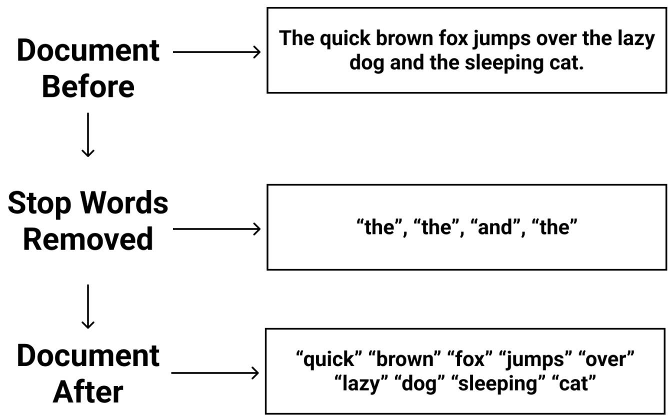


Figure 2.5: Stop Word Removal - Removing unimportant words

jumped over the lazy dog.". The word "brown" is an adjective.

Conversely, in the sentence "Mr. Brown jumps over the lazy dog,"; the word "Brown" is a proper noun. This example illustrates the importance of neighbour words. Brown is contained in both sentences, but the preceding words ("The quick," "Mr.") change the part of the speech. POS tagging is essential for building lemmatizers [4].

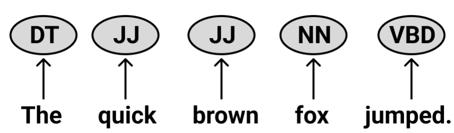


Figure 2.6: Parts of Speech - Tagging words with their part of speech

Lemmatization

Reducing the total surface area of the dataset by reducing word count, removing outliers and clustering terms more closely together can be achieved by lemmatization. Lemmatization is a process in which we group words together based on a root. For example Figure 2.7, the words “jumps,” “jumped,” and “jumping” all have the same root word, jump. Therefore, we can reduce common words to a root utilizing the POS tags to preserve context.

trying tried try	⟶	try
talking talked talk	⟶	talk
running ran run	⟶	run

Figure 2.7: Lemmatization - Reducing words to their root

Input Transformation

Preprocessed language text is the first step in training a language model. To validate the data for a machine learning model, it is necessary to transform the data into something computationally compatible. Input transformation is a process by which we convert the text-based features into a numerical vector space model. Two standard methods of achieving a transformation are bag-of-words BOW or TF-IDF.

BOW

BOW is a simple order less vector document representation model as shown in Table 2.2 where word counts per document are stored. The word counts can then be utilized with machine learning models. For example, take the following sentences represented as individual short documents not considering the previous preprocessing steps discussed in the previous section:

- Document 1: "The quick brown fox. . . ."
- Document 2: "The slow brown frog. . . ."
- Document 3: "The big red dog. . . "

	The	Quick	Brown	Fox	Slow	Green	Frog	Big	Red
Document 1	1	1	1	1	0	0	0	0	0
Document 2	1	0	1	0	1	1	1	0	0
Document 3	1	0	0	0	0	0	0	1	1

Table 2.2: Bag of Words Model

Table 2.2 demonstrates how BOW would be logically created in that each document is transformed into a vector containing a count of the words within that document. The main drawback of this method is that we must consider the words that appear in the documents. The word "frog" is only seen in the "Document 2" vector; however, a 0 count must be kept for each word in the other document vectors. This can result in large, unwieldy vectors when considering a large Corpus with substantial amounts of text. In addition to large vectors, it must be noted that the context and word ordering of the document is lost with a BOW transformation.

TF-IDF

An alternative to the BOW is the TF-IDF. Rather than storing word counts, the frequency of terms that appear in a specific document is considered. Word frequency in each document is compared and scored against the frequency of the word appearing across all documents. If a word rarely appears in many documents, it is given more weight than a word frequently in many documents. Given our previous example, a TF-IDF representation of the documents could look something like Table 2.3.

	The	Quick	Brown	Fox	Slow	Green	Frog	Big	Red
Document 1	0	0.48	0.25	0.48	0	0	0	0	0
Document 2	0	0	0.25	0	0.48	0.48	0.48	0	0
Document 3	0	0	0	0	0	0	0	0.48	0.48

Table 2.3: Term Frequency Inverse Document Frequency

N-Grams

An n-gram is a slice of text or group of words. For example, in the following sentence, "The quick brown fox jumps over the lazy dog." An Ngram of 1-1 would result in text transformation to the following ["The", "quick", "brown", "fox",...]; this is also known as unigrams. It is also common to use bigram and trigrams as shown in Figure 2.8.

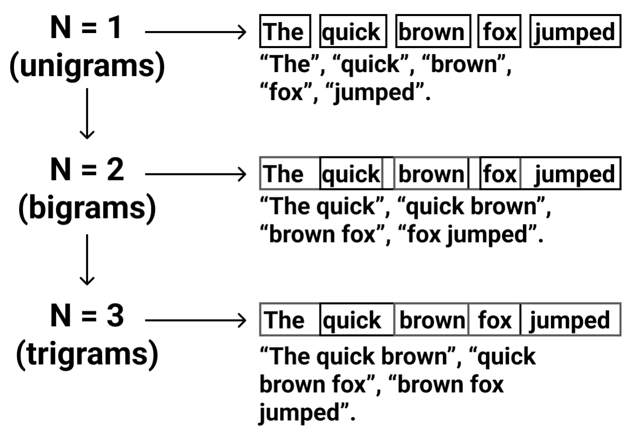


Figure 2.8: 1,2, and 3 N-Gram examples

Performance

Algorithmic performance is typically measured in accuracy or via an F1 Score. Accuracy is a measure to determine the correctly identified data points over all possible predictions. F1 is a weighted measure that also considers incorrectly classified data. F1 uses precision, recall and can consider data distribution when the data is imbalanced. Figure 2.9 and Figure 2.10 demonstrates the formulas for both accuracy and F1.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

Figure 2.9: Accuracy Measure Formula

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Figure 2.10: F1 Measure Formula

Summary

To summarize the important takeaways for this chapter include sufficient background material relating to the Canadian court system as well as the judiciary and how a judge may operate with judicial independence. This results in case law known as precedent that in effect is law. Future decisions in a courtroom are bound by precedent and must follow suit if they are similar. Given precedent case law as a dataset , authors such as [36, 38, 3, 52, 59, 46, 35, 25, 32, 37, 48, 27] have shown some merit to utilizing the techniques described in chapter 3 to classify future cases above 80 f1 score.

Chapter 3

Methodology

3.1 Research Purpose

This study aims to investigate and compare the classification performance of various machine learning algorithms with court cases from the Canadian Supreme Court as a dataset. This research will show the performance of varying algorithms, namely KNN, NB, LR, SVM, and their ability to determine the outcome of pre-existing court cases. The result of this experiment is a preliminary step in a more extensive scoped study to explore court case prediction that will be described further in future work chapter 6.

3.2 Implementation Design

Training any machine learning algorithms requires a dataset. The case and outcomes from the SCC will be a dataset for this experiment. Currently, SCC cases are available through the SCC website [11] or a third-party website, CanLII [1]. Neither site offers unfettered access for bulk download or machine-readable formats. This dataset needs to be constructed semi-manually before applying machine learning algorithms found in the literature. The steps involved in the methodology of the implementation are found in figure 2.3.

Data Acquisition

The SCC judgements database has more than 11,000 documents from 1877 to 2021. Given the sheer volume of cases, the decision was made to obtain permission from the SCC before accessing the documents. The Director of the Library Branch of the SCC for education and research under the Reproduction of Federal Law Order permitted access to the documents. Permission was given under one strict guideline: "the SCC nor Lexum has the capacity to provide bulk access to the judgements. SCC Judgments may be downloaded manually from the site, in small batches . . . ." See redacted email correspondence permission in Appendix A

Dataset Construction

The dataset in question requires construction, as mentioned in section 3.2 and section 3.2.1, as there is no freely accessible machine-readable dataset for court cases from the SCC. First, it was necessary to download the court cases manually to construct the dataset. The manual acquisition of the files occurred multiple times per day over multiple sessions. According to the instructions provided by the SCC found in Appendix A, it was necessary to slowly download the cases to avoid triggering automated protections built into the website. Over a few weeks, roughly 30-60 cases were downloaded 3-4 times daily. Two file formats were available for download, PDF, and Word document format (.docx, .doc, and .wpd). The word document format was chosen over PDF as many older PDFs contained only scans, whereas the word documents contained texts that allowed for text extraction methods.

Initial Dataset Exclusions

Processing the Word files with varying formats leads to complexities not foreseen during the data acquisition stage. The differences in the structure of the varying Word document types mean that it was not feasible to modify data extraction functions to be compatible with document types. Excluding documents before 2011 ensured that every document would be in a consistent format (.docx). Given more time, it would be feasible to write multiple data extraction functions; this is discussed further in the chapter 5 and chapter 6. For a comprehensive list of the case exclusions for the years utilized, see Appendix C.

Metadata Extraction

Each court case has two sets of metadata. The first set is internal metadata consisting of data about the file—such as word counts, line counts, characters, and paragraph counts. The values of the internal metadata ensure that the selected cases have a similar structure. Figure 3.1 and Figure 3.2 contain average case metadata such as page and word count.

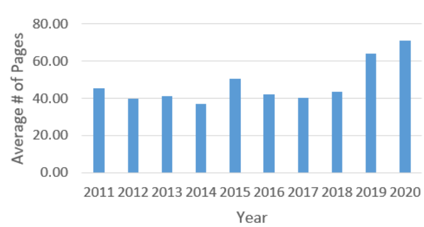


Figure 3.1: Average number of case documents per year

This process found that not every case was the same and some were not fit for

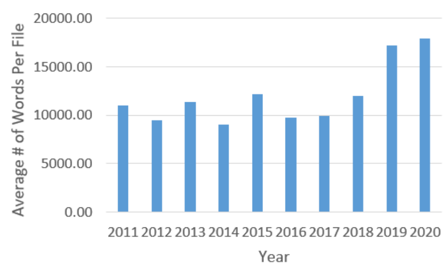


Figure 3.2: Average words per document by year

inclusion. Excluded documents include those containing less than ten pages and those containing multiple verdicts since they had multiple (trials/cases/applicants or defendants/issues). Figure 3.3 shows the number of exclusions for a given year; on average, roughly 29% of cases were excluded for one or more reasons mentioned above. A complete list of exclusions can be found in Appendix C.

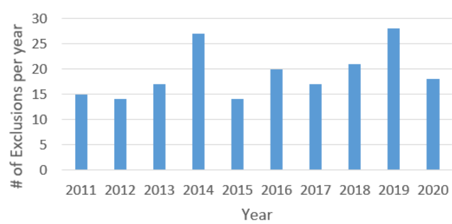


Figure 3.3: Number of document exclusions per year

The second set of metadata is related to the file's content, meaning the court data. The content metadata includes the Citation, Appeal Date, Judgement Rendered Date, Docket Number, and Outcome (appeal allowed or appeal dismissed). Again, this metadata was helpful for further exclusions for cases that were unfit for training; as previously mentioned, the completed list of exclusions is available in Appendix C. After all exclusions are applied, there is a remainder of 436 cases. The distribution of cases resulting in an “allow” verdict and a “dismiss” verdict can be seen in Figure 3.4.

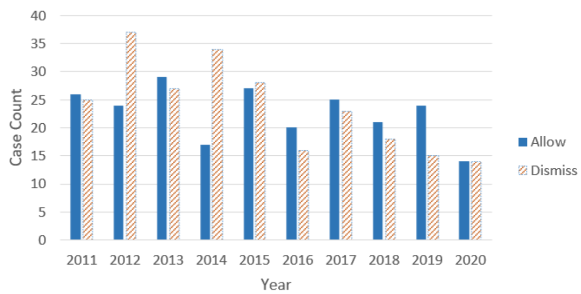


Figure 3.4: Court Case Distribution: Allow vs Dismiss

The internal metadata extraction method required conversion from the .docx format to a zip folder. Once converted to a zip folder, it was possible to open the folder utilizing the python library 'zip file'. The next step was to programmatically access the "docProps/app.xml" file to retrieve the abovementioned metadata. The zip folder was then closed and converted back to .docx format. All metadata was stored in a CSV format for later processing. The following table (Table 3.1) shows the structure of the CSV format and how the data was stored.

Table 3.1: Metadata extracted from case documents

Index	File	Pages	Words	Characters	Lines	Paragraphs	Citation	Appeal Heard	Judgement Rendered	Docket
0	AG_Ontario-en.docx	173	42970	228125	1901	541	Ontario (Attorney General) v. G, 2020 SCC 38	20-Feb-20	20-Nov-20	38585

The content metadata was acquired using the 'docx', 'os', and 'pandas'python libraries that allowed text extraction. The metadata extraction was performed using regular expressions once the text was isolated from the file. Typically, a title preceded each variable value. For example, the citation might include: "Citation: SCC[145, 2020] ". A regular expression would look for the word "Citation:" and save the text immediately after, for example, SCC[145, 2020]. Content metadata was appended to the CSV file mentioned earlier.

In some cases, the metadata extraction process would not be complete, and some metadata variables were left empty. The errors were due to inconsistencies in the file formatting. The empty or missing values were corrected manually.

Court Data Extraction

This final step in the dataset construction is to copy the source data from the .docx files and append it to the same CSV file mentioned above. The entirety of the document was appended to the last column of each court case. The result is a singular CSV file where each row contains both the metadata and case data.

Pre-Processing

As discussed in section 2.1, preprocessing the data is a vital step in text-based machine learning problems. Human language is exceptionally complex for algorithms to understand without preprocessing performed on the text. The preprocessing step allows us to remove noise (remove repetition, complex sentences, passive writing, and nominalization section 2.2), thereby shrinking the dataset, removing unimportant words and reducing lengthier words to their root word equivalent. There are no requirements for preprocessing steps, but recommendations and conventions that should be tested on each dataset to find optimum steps. The preprocessing steps taken in this experiment are similar to those undertaken by [36, 38, 3, 52, 59, 46, 35, 25, 32, 37, 48]. The preprocessing steps performed are as outlined in the NLP pipeline in section 2.3. The text from each case was extracted and saved in individual text files. This step immediately removed any formatting from the documents. Pre-processing then occurred for each file using the following methods: Tokenization, Parts of Speech Tagging, Lemmatization and Stop word removal, as described in section 2.3. A determination occurred to perform all the preprocessing steps as court cases tend to have much noise in terms of excess words as an objective to be precise in the court of law. Reducing noise in the data should lead to more closely clustered essential terms, which should benefit the overall performance (section 2.1).

3.3 Limitations

The limitations of this study reside with the SCC dataset.[11] There are two main issues relating to the SCC court cases:

1. The SCC judgements make up the dataset for this study. The cases are not available for bulk download and must be acquired manually. There exist 11000+ documents ranging from 1877 to date.

2. The second issue relates to the computability of the court cases. There are only two download options available for the cases. The first format is pdf files that are often only scanned and do not contain text. The second format is word-based, and the file extensions vary between '.docx','.doc', and '.wpd'. Due to the time constraints of the study, it will only be feasible to include data from 2011 to 2019. These limitations are

discussed in further detail in section 5.2.

3.4 Training Phase

The literature on NLP and law applies diverging strategies when choosing training methods. For simplicity and time constraints, four of the most common reoccurring machine learning algorithms were selected from the literature **(“SVM”, “LR”, “KNN”, and “NB”)**. The training models and methods were derived from SK Learn Libraries [4] and the preprocessed data from section 3.2. Before training, it is necessary to split the data into two distinct groups: testing and training data. The model selection algorithm from the sklearn python package facilitates this process (Figure 3.5). Table 3.2 defines the variables and data used in the model selection.

```
Train_X, Test_X, Train_Y, Test_Y = model_selection.train_test_split(
     (data['text'],data['Outcome']),
     (test_size = ‘TestSize’),
     (random_state=42)
);
```

Figure 3.5: Scikit Model Selection: Train Test Split

Following the model selection, it was then necessary to vectorize the data using the TF-IDF sklearn method (Figure 3.6 and Table 3.3). This method is also known as feature extraction. It effectively turns the text into a machine-readable matrix, as presented in section 2.3. Additionally, MaxDF is considered during the vectorization process. During this process, term frequency is calculated and should the frequency exceed the maximum threshold, then the term is ignored. MaxDF is described further in Table 3.2.

Table 3.2: Training Parameters

Parameter	Description
data['text']	refers to the entirety of the dataset
data['outcome']	the training labels associated with the dataset
test-size	was investigated to determine if varying testing/training ratios impacted performance
random_state	This paramater controls the shuffling of the dataset prior to splitting the dataset. A random state was used for replicability, in that running in the same data with the same parameters and the same random state would result in the same output [4].

```
Tfidf_vect = TfidfVectorizer(
     (max_features = [max_df]),
     (encoding = 'utf-8'),
     (ngram_range = (1, 2))
);
```

Figure 3.6: Scikit TFIDF Vectorizer initialization

Table 3.3: N-Gram and Max Document Frequency

Parameter	Description
NgramRange	The Ngram Range parameter allows for groupings of terms to preserve context. NGRAMs are described in detail in Section 2.6.9. Multiple n-gram ranges were investigated for overall training impact and are discussed further in Section 4.
MaxDF	The Max Document Frequency parameter is used to ignore frequent terms that appear with a given frequency within a document. Max DF is typically used to remove terms that appear too frequently in a document. Frequently appearing words could be considered corpus-specific stop words. This method allows for excluding such words without taking inventory of each word of the corpus and evaluating its importance. For example, a Max DF of 0.5 will remove words that appear in more than 50% of the documents. Varying MaxDF settings were used to test the effects against the dataset.

Completion of the data vectorization leaves the final task of training and testing. Initialization of the classification model is supervised; the labels are trained on a testing set before the final predictions occur. Once the predictions take place, the outputs are recorded for evaluation. This process is repeated for each algorithm. The results are available in chapter 4.

Summary

In closing the methodology included various stages as shown in section 2.3. It was first necessary to collect the raw case data. Extract some necessary metadata such as training labels. Following this the data was perpocessed using general NLP techniques before finally being transformed with the Scikit TFIDF vectorizer. During the transformation various maximum document frequency parameters were tested (0.3, 0.5, 0.7). This means that a certain defined threshold of documents were removed from the data during vertorization. Finally the vectorized outputs were trained on ML algorithms such as SVM, NB, LR, KNN. Limitations to the experiment resulted in document exclusion due to document formatting and inconsistencies in document content as described in section 3.3.

Chapter 4

Results

For the experiment, data were collected from 144 variations of the four algorithms discussed in section 3.1 and section 3.4. The data collected can be categorized and separated by the algorithm and parameter adjustment variations between runs. The results were processed utilizing excel and excel formulas.

4.1 Overview of Results

The total court case count after exclusions (See Appendix C for a complete list of exclusions) is 436, where 223 cases had the outcome of 'dismiss' and 213 had the outcome of 'allow'; the outcome decisions were removed from the case data prior to training. The data collected from the implementation includes parameters such as testing size, n-gram range, and MaxDF. As mentioned, varying parameters were tested to study change effects on the performance of the algorithms. Table 4.1 outlines the parameter settings, and each row consists of settings for an individual run. Each run was repeated for each algorithm. A complete list of all testing runs is available in Appendix B. It was decided not to shuffle the training-testing data and dismiss performing cross-validation due to court case precedence. Precedence is described in Section section 2.2 and creates order regarding court case outcome, as older cases may be cited in new cases. As new decisions are made, the law will evolve accordingly. Therefore, shuffling steps were omitted in order to maintain their historical order.

In machine learning implementations, measuring performance via precision, recall, and F1 Score is generally accepted. In this instance, precision is a percentage of positive 'dismissed' cases that have been identified correctly. Recall is a percentage of cases that have been correctly identified as being "dismissed" divided by the total classifications for "dismissed". F1 is a combination of precision and recall, the formulas for each of these metrics are available in figures 4.1, 4.2, 4.3, and 4.4.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

Figure 4.1: Precision Formula

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

Figure 4.2: Recall Formula

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{4.3}$$

Figure 4.3: Accuracy Formula

Table 4.1: Iterations parameter settings tested for each algorithm trained

Test Size	Ngram Low	Ngram High	MaxDF
0.1	1	1	0.3
0.1	1	1	0.5
0.1	1	1	0.7
...	...	...	...
0.4	3	3	0.3
0.4	3	3	0.5
0.4	3	3	0.7

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \qquad (4.4)$$

Figure 4.4: F1 Formula

Table 4.2 is an overview of the results for the 144 runs. Table 4.3 shows that the allow was labelled as 0 and dismiss as 1. Regarding the maximum performing algorithm, it was found that in terms of F1 Score, NB had the maximum overall performance with an F1 Score of 61. Parameters for this run included a testing size of 0.4 and bigrams only. It was found that MaxDF did not contribute to significant changes in the overall F1 Score. The F1 Score on average across all runs was 51.8. The lowest-performing algorithm was KNN, with an F1 score of 36.36; parameters for this run included a testing size of 0.1, unigrams only, and a MaxDF of 0.3.

Table 4.2: Overview of Results

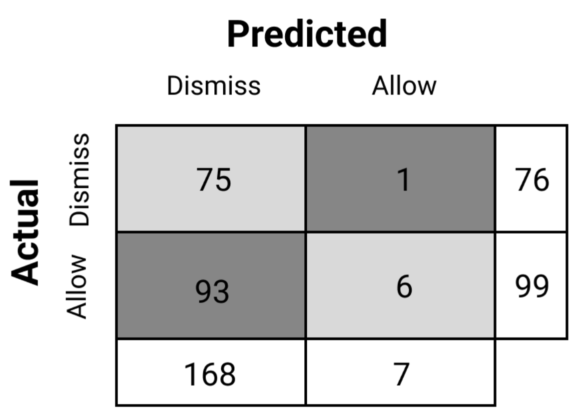
Top Result (Test Size)	**KNN**	**LR**	**NB**	**SVM**
0.1	46.5116	48.1481	47.0588	47.8261
0.2	53.9326	56.6667	54.86773	54.5454
0.3	56.7276	59.4595	59.8802	56.37584
0.4	56.7308	61.2903	61.4754	57.8947
Worst Result (Test Size)				
0.1	36.3636	40	43.1372	39.1304
0.2	37.3626	49.5238	50.8772	46.9388
0.3	44.2857	54.1935	55.3459	48.1752
0.4	48.1675	57.7778	59.6491	50
Average Result (Test Size)				
0.1	41.4113	44.8497	44.5919	42.6511
0.2	48.5005	53.8703	53.1710	52.2939
0.3	51.2960	57.7576	56.8941	53.2869
0.4	53.0334	59.8589	60.7495	54.8846
Best Overall	56.7376	61.2903	61.4754	57.8947
Worst Overall	36.3636	40	43.1372	39.1304
Average Overall	48.5603	54.0841	53.8516	50.7791

Figure 4.5 shows the confusion matrix for the highest-ranking F1 Score, the algorithm being NB. The classification algorithm was more likely to classify a case as ‘dismiss’ rather than ‘allow’. This pattern generally repeated for each algorithm, and there seemed to be a bias towards the ‘dismiss’ label. The bias is discussed further in section 5.3.

Table 4.3: Label Encoding

Allow	0
Dismiss	1

Figure 4.5: Maximum Scoring Confusion Matrix



4.2 Algorithm Results

The following section discusses the results of the following performance measures; F1 Score, Accuracy, Precision, and Recall.

F1 Results

Table 4.4 illustrates the minimum, maximum and average F1 Score per algorithm. KNN had *minimum* performance for individual runs with an *F1 Score of 36.36*; *KNN* also *performed minimally* on average with an *average F1 score of 49*. *NB* had the *maximum* overall performance with an *F1 score of around 61;* on average, NB and LR regression performed similarly, with F1 Scores around 54.

Table 4.4: F1 Score Results

Algorithm	Min	Max	Avg
NB	43.1372549	61.47540984	53.85164039
SVM	39.13043478	57.89473684	50.77911228
KNN	36.36363636	56.73758865	48.56031212
LR	40	61.29032258	54.08413882

Accuracy Results

Table 4.5 shows the Accuracy achieved per algorithm. The accuracy measure is as described in section 2.1. *SVM had minimum performance* overall with an *Accuracy score of 4.4*. The *maximum* performing accuracy was*SVM with an* *accuracy score of 54.54.* On average SVM and KNN performed similarly, with an Accuracy score of 45-46.

Table 4.5: Accuracy Score Results

Algorithm	Min	Max	Avg
NB	31.81818	48.85496	41.220256
SVM	4.411786	54.54545	45.20978236
KNN	35.22727	53.43511	46.14413354
LR	31.81818	50.38168	41.75705529

Precision Results

Table 4.6 shows the results of the precision per algorithm. In this case, the minimum-performing algorithm was KNN. In contrast, the maximum performing algorithm in terms of precision is KNN. On average, all the algorithms performed similarly, scoring 39-40.

Table 4.6: Precision Score Results

Algorithm	Min	Max	Avg
NB	31.42857143	45.45455	39.64005
SVM	30	46.15385	40.63515
KNN	28.57142857	47.61905	39.91987
LR	29.41176471	45.91837	39.8899

Recall Results

Table 4.7 shows the results of the recall score per algorithm. On average, NB and LR performed similarly, with an average recall score of 84. LR had maximum recall performance overall with a score of 100, followed closely by NB with a score of 98. Finally, the minimum recall scores were KNN, followed closely by SVM.

Table 4.7: Recall Score Results

Algorithm	Min	Max	Avg
NB	68.75	98.68421	84.19022
SVM	50	86.84211	68.1713
KNN	47.22222	77.63158	62.22385
LR	62.5	100	84.42272

These results demonstrate that the machine learning algorithms could still detect some patterns even with less than optimal performance. In more than just a few cases, the algorithms obtained a performance more significant than a 50-50 guess where F1 scores above 50 on a balanced dataset denote more outstanding performance than a human participant making random classification guesses. This can be seen in the F1 scores achieved for both NB and LR; as previously mentioned, achieving scores of 60, at least 10 points above the acceptable baseline. Regardless of performance, it is clear that the F1 Score was largely unaffected by the Max DF parameter. A Testing Size of 0.4 and trigrams, on average, performed better. Table 4.8 shows the top 25% of runs and illustrates the higher performance of both testing sizes and trigrams. Figure 4.6 However, more work is required to fine-tune the Pipeline as described in section 2.1.

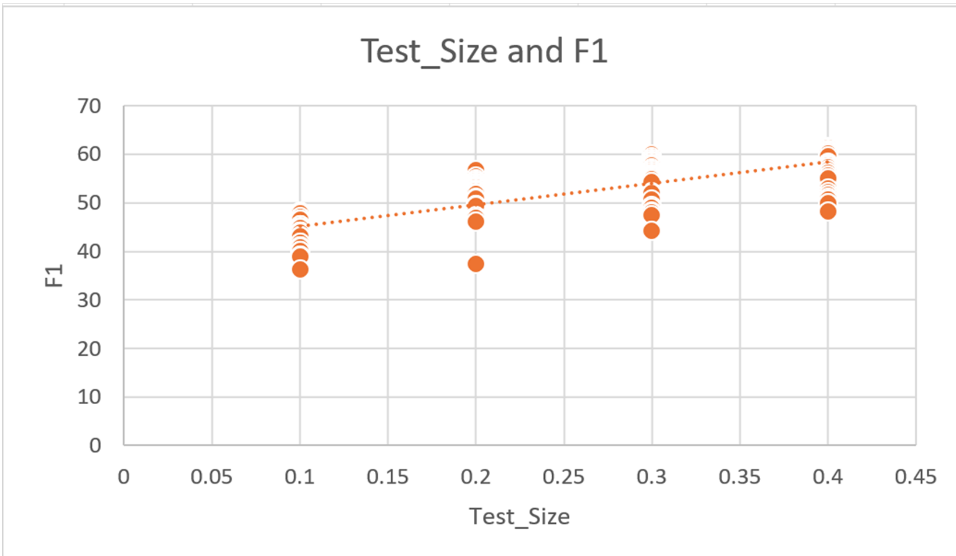


Figure 4.6: Test Size results over F1

Table 4.8: Results of the Top 35 performers

Algorithm	Occurences	MaxDF		NGram		Test Size	
NB	14	0.3	4	Unigrams	5	0.1	0
		0.5	5	Bigrams	3	0.2	0
		0.7	5	Trigrams	6	0.3	5
						0.4	9
LR	16	0.3	5	Unigrams	4	0.1	0
		0.5	5	Bigrams	6	0.2	0
		0.7	6	Trigrams	6	0.3	7
						0.4	9
SVM	3	0.3	1	Unigrams	0	0.1	0
		0.5	1	Bigrams	0	0.2	0
		0.7	1	Trigrams	3	0.3	0
						0.4	3
KNN	2	0.3	0	Unigrams	0	0.1	0
		0.5	1	Bigrams	0	0.2	0
		0.7	1	Trigrams	2	0.3	2
						0.4	0
Totals	35	0.3	10	Unigrams	9	0.1	0
		0.5	12	Bigrams	9	0.2	0
		0.7	13	Trigrams	17	0.3	14
						0.4	21

Chapter 5

Discussion

In this chapter, the results of the algorithm runs will be discussed, corresponding to the hypothesis questions presented in section 3.3. Comparisons will also be drawn to relevant literature that conducted similar work.

5.1 Hypothesis

R.Q1 Can we classify outcomes of the Supreme Court of Canada with some degree of Accuracy above a 50-50 guess (above 50% F1 Score)?

Table 4.4 results show that multiple algorithms with varying parameters can achieve a greater than 50% F1 score for this dataset. On average, the algorithm's F1 performance was slightly higher than 50%, with some of the best overall performances in the low 60s. With this information, improvements to the NLP pipeline could result in a higher F1 score (above 60%). Discussions relating to the steps that can be taken to achieve higher performance are discussed in chapter 6. Regardless, classification of the SCC cases is possible given the 60% F1 score.

R.Q2 Which algorithm obtains the best performance for this dataset?

It was hypothesized that SVM would have the most remarkable performance as this was a trend in the literature review [36, 38, 3, 52, 59, 46, 35, 25, 32, 37, 48]. However, in this experiment, SVM was not the highest performing algorithm. Overall performance is generally measured by utilizing an F1 Score. NB achieved the highest F1 Score 61 with a testing size of 0.4, bi-grams only, and a MaxDF of 0.3. Additionally, the average F1 Score of LR was greater than that of NB, which on average, obtained an average F1 Score of 54.

R.Q3 Which training-testing ratio provides the best performance for this data set?

Figure 4.6 and Table 4.8 confirms that higher testing ratios on average relate to higher performance. A testing size of 0.4 typically resulted in better performance than lower test sizes. Table 4.8 also shows that tri-grams, on average, were more likely to perform better. Future versions of this work should include a more significant subset of the overall data, a greater variance in test size parameter values, and combinations of n-gram range to better understand the performance effects.

5.2 Limitations

This experiment had several limitations, including data and parameter/library limitations. These limitations are mainly due to the court cases' size, complexity, and data format because a complete and computable dataset for court cases from the SCC does not exist. Therefore, this required the construction of the dataset from scratch (see section 3.2). Limitations to these steps are elaborated on further.

The dataset constructed consisted of SCC court cases from 2011 to 2019. Including as much data as possible with any machine learning dataset would be ideal but not feasible given the time confines. It would have been preferable to include more case data prior to 2011. For example, from 1954 to 2011, these 57 years contained over 3,300 documents without removing exclusions. Therefore, around 3,300 documents could have been added to this dataset. Although, cases prior to 2011 are stored in varying formats, including '.wpd', '.doc', and '.docx'. The different case formats meant that general text extraction methods would not be adequate. File layouts such as headings and spacings are also different in the different file formats. More advanced or manual methods would be required to include this data. Unfortunately, that was not reasonable given the time confines of the experiment. Future iterations of this work should include this data as described in chapter 6.

Additionally, extracting further case information of the specific court cases, such as legal groupings or sitting judges, could have been helpful during the training/testing of these algorithms to provide further insights into the data. For example, grouping the cases by area of law may help to improve performance by only training algorithms with similar case groupings (family law, criminal law, constitutional law). These groupings could be helpful because word selection could drastically differ between criminal, constitutional or family law cases. Unfortunately, there does not exist a computer-readable source of case types that could have been leveraged for this task; this information would need manual intervention to be added. Finally, information regarding sitting Judges and their voting history could also be a valuable mechanism for court case outcome forecasting combined with the additional metadata mentioned in this section.

The second limitation rests with the algorithm parameters and libraries used. The SKLearn Python Library was used for simplicity and consistency as there were methods for conducting each of the selected algorithms. However, it may be prudent to repeat the same experiment with different machine learning libraries and compare the results. Similarly, some parameters were tested with method parameters to understand the effects of different values on the algorithm performance. Additional parameters in SKlearn and other libraries can be modified to test performance. Additional parameters and pipeline considerations are described further in chapter 6.

5.3 Label Bias

Given the results described in chapter 4, the algorithms have produced a bias during their training as they tend to lean towards a 'dismiss' classification. It is not uncommon for bias towards the majority class to occur when training on unbalanced datasets. However, it should be noted that cases labelled as 'dismiss' contributes to 51% of the total cases and 'allow' consist of 49% of the total cases. [29] states that typically insignificant class imbalance ratio could be as low as 1:4, whereas the ratio for included court cases is less than 1:4 at a 51:49. Given the small ratio, it was not necessary to address the imbalance via the Synthetic Minority Oversampling Technique (SMOTE) technique [29].

The bias is unclear and should be investigated further; this is discussed in chapter 6.

5.4 Summary

Before we can confidently use AI in the courtroom more discussion around various ethical and societal topics are required. However, it has been shown that the case data from the scc has potential for the basis of training data for such systems. There are limitations to this data that need to be overcome in future iterations of this work. It must also be noted that the label bias should be investigated further as the algorithms tended towards a dismiss classification. Optioms to pursure corrections of this bias are discussed further in chapter 6.

Chapter 6

Conclusion and Future Work

From the results produced by the experiment, it is clear that the classification of SCC court cases has viability. The highest performing model correctly classified a court case about 61% of the time; this is undeniably greater than a 50% guess baseline. Unlike other researchers with a more significant performance with the SVM algorithm, it is still unclear why NB performed better in this experiment. These results are a confirmation that this type of data could be used for Categorization in addition to forecasting. However, for a machine learning model to be considered a predicting model on court cases, it must only be trained with data that exists prior to the final SCC outcome.

Future iterations of this work should include a larger subset of the available data. Care should be taken to ensure that the extraction methods are consistent, given the varying file formats and layouts. It has previously been discussed that there are no set steps for NLP techniques; instead, there are recommendations. It would be prudent, however, to test not only additional algorithms but additional parameters such as:

- Combinations of n-grams ([unigrams, and bi-grams], [unigrams and tr-grams],[bi-grams, and trigrams]): N-grams help to maintain context, which may be necessary for court case wording.
- Maximum and Minimum document frequencies: Minimizing frequently occurring and rare words with these parameters will help reduce corpus-specific stop words. Legal language tends to have additional wording that may be unnecessary for language models.
- Attempt to classify or predict cases based on year or area of law: Grouping the dataset by area of law may help to improve performance as case similarity will be more closely aligned.

Additionally, more work should be done to investigate the origin of the label bias in the models. As previously mentioned, the bias does not originate from dataset imbalance but some other unknown source. Improving data extraction and cleansing techniques combined with the grouping methods mentioned above may help alleviate some bias in the model. The scope and continuation of this additional work could be used as the prerequisite fulfilment of a Ph.D. degree.

As mentioned, the groundwork presented in this thesis could be extended into industry. However, prior to any industry marketability, it is essential to further discussions around the pros and cons of automated systems in law as well as the ethical implications and societal impacts of using such technology in law. We must also determine the threshold of acceptability of society concerning some measurable outcome, I.e. would society accept an automated system with an F1 Score of 90%, or is the threshold perhaps 98%?

Alternatively, is there some other measurable metric threshold we must surpass before society widely accepts such technology?

Bibliography

[1] URL: https://www.canlii.org/en.

[2] Benjamin Alarie, Anthony Niblett, and Albert H Yoon. "How artificial intelligence will affect the practice of law". en. In: *Univ. Tor. Law J.* 68.supplement 1 (Jan. 2018), pp. 106–124.

[3] Nikolaos Aletras et al. "Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective". In: *PeerJ Computer Science* 2 (2016), e93.

[4] *API reference*. URL: https://scikit-learn.org/stable/modules/classes.html.

[5] Theo Araujo et al. "In AI we trust? Perceptions about automated decision-making by artificial intelligence". en. In: *AI Soc.* 35.3 (Sept. 2020), pp. 611–623.

[6] Theo Araujo et al. "In AI we trust? Perceptions about automated decision-making by artificial intelligence". In: *AI & SOCIETY* 35.3 (2020), pp. 611–623.

[7] Katie Atkinson, Trevor Bench-Capon, and Danushka Bollegala. "Explanation in AI and law: Past, present and future". In: *Artificial Intelligence* 289 (2020), p. 103387. ISSN: 0004-3702. DOI:

https://doi.org/10.1016/j.artint.2020.103387. URL: https://www.sciencedirect.com/science/article/pii/S0004370220301375.

[8] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with python: Analyzing text with the natural language toolkit.* O'Reilly, 2009.

[9] A Boer, R Winkels, and F Vitali. "Proposed XML standards for law: Metalex and LKIF". In: *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS* 165 (2007).

[10] Dale Brawn. *Paths to the Bench: The Judicial Appointment Process in Manitoba, 1870-1950.* UBC Press, 2014.

[11] Supreme Court of Canada. *SCC case information - search.* 2012. URL: https://decisions.scc-csc.ca/scc-csc/scc-csc/en/nav_date.do.

[12] D Cavar, J Herring, and A Meyer. "Case law analysis using deep NLP and knowledge graphs". In: *Proceedings of the LREC.* 2018.

[13] Steve Coughlan. "Canadian law dictionary". In: (2013).

[14] Arthur Dyevre. "Text-mining for lawyers: How machine learning techniques can advance our understanding of legal discourse". In: *Erasmus Law Rev.* 14.1 (Oct. 2021).

[15] Arthur Dyevre. "The promise and pitfall of automated text-scaling techniques for the analysis of jurisprudential change". en. In: *Artif. Intell. Law* 29.2 (June 2021), pp. 239–269.

[16] Lance Eliot. "Authorized and Unauthorized Practices of law: The role of autonomous levels of AI Legal Reasoning". In: (Aug. 2020). arXiv: 2008.09507 [cs.CY].

[17] Amitai Etzioni and Oren Etzioni. "Incorporating ethics into artificial intelligence". en. In: *J. Ethics* 21.4 (Dec. 2017), pp. 403–418.

[18] F Fagan. "Natural language processing for lawyers and judges. Mich". In: *L. Rev* 119 (2020).

[19] Kranti Vithal Ghag and Ketan Shah. "Comparative analysis of effect of stopwords removal on sentiment classification". In: *2015 International Conference on Computer, Communication and Control (IC4)*. 2015, pp. 1–6. DOI: 10.1109/IC4.2015.7375527.

[20] Department of Justice Government of Canada. *Canadas Court System*. 2015. URL: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjd9KWBnun4AhWoIOQIHX4uBNQQFnoECAcQAQ&url=https%3A%2F%2Fwww.justice.gc.ca%2Feng%2Fcsj-sjc%2Fccs-ajc%2Fpdf%2Fcourten.pdf&usg=AOvVaw2VR36no3jQQYT4QwExswKg.

[21] Department of Justice Government of Canada. *The appeal process in Canada*. Sept. 2021. URL: https://www.justice.gc.ca/eng/csj-sjc/just/appeal-appel.html.

[22] Thilo Hagendorff. "The ethics of AI ethics: An evaluation of guidelines". en. In: *Minds Mach. (Dordr.)* 30.1 (Mar. 2020), pp. 99–120.

[23] Hendrik Heuer and Daniel Buschek. "Methods for the Design and Evaluation of HCI+NLP Systems". In: *CoRR* abs/2102.13461 (2021). arXiv: 2102.13461. URL: https://arxiv.org/abs/2102.13461.

[24] Mireille Hildebrandt. "Law as computation in the era of artificial legal intelligence: Speaking law to the power of statistics". en. In: *Univ. Tor. Law J.* 68.supplement 1 (Jan. 2018), pp. 12–35.

[25] Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. "Summarization of legal documents: Where are we now and the way forward". In: *Computer Science Review* 40 (2021), p. 100388. ISSN:

1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2021.100388. URL: https://www.sciencedirect.com/science/article/pii/S1574013721000289.

[26] V Kaliazin and N Kaliazina. “Limitations of Leveraging ai technologies for court decisions prediction”. In: *The Scientific Heritage* (2019), pp. 55–58.

[27] Daniel Martin Katz, Michael J Bommarito, and Josh Blackman. “A general approach for predicting the behavior of the Supreme Court of the United States”. In: *PloS one* 12.4 (2017), e0174698.

[28] Mark S Krass. *Learning the Rulebook: Challenges Facing NLP in Legal Contexts*. 2019.

[29] Bartosz Krawczyk. “Learning from imbalanced data: open challenges and future directions”. In: *Progress in Artificial Intelligence* 5.4 (2016), pp. 221–232.

[30] Kevin LaRoche, M Laurentius Marais, and David Salter. “The Length of Civil Trials and Time to Judgment in Canada: A Case for Time-Limited Trials”. In: *Can. B. Rev.* 99 (2021), p. 286.

[31] Migle Laukyte. “AI as a Legal Person”. In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. ICAIL ’19. Montreal, QC, Canada: Association for Computing Machinery, 2019, pp. 209–213. ISBN: 9781450367547. DOI: 10.1145/3322640.3326701. URL: https://doi.org/10.1145/3322640.3326701.

[32] Zhenyu Liu and Huanhuan Chen. “A predictive performance comparison of machine learning models for judicial cases”. In: *2017 IEEE*

Symposium Series on Computational Intelligence (SSCI). 2017, pp. 1–6. DOI: 10.1109/SSCI.2017.8285436.

[33] Marko Marković and Stevan Gostojić. "Open Judicial Data: A Comparative Analysis". In: *Social Science Computer Review* 38.3 (2020), pp. 295–314. DOI: 10.1177/0894439318770744. eprint: https://doi.org/10.1177/0894439318770744. URL: https://doi.org/10.1177/0894439318770744.

[34] J Mcgill, S Bouclin, and A Salyzyn. *Mobile and Web-based Legal Apps: Opportunities, Risks and Information Gaps. Risks and Information Gaps*. 2017.

[35] Juan Carlos Medina Serrano, Orestis Papakyriakopoulos, and Simon Hegelich. "NLP-based Feature Extraction for the Detection of COVID-19 Misinformation Videos on YouTube". In: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Online: Association for Computational Linguistics, July 2020. URL: https://aclanthology.org/2020.nlpcovid19-acl.17.

[36] M Medvedeva, M Vols, and M Wieling. "Judicial decisions of the European Court of Human Rights: Looking into the crystal ball". In: *Proceedings of the conference on empirical legal studies*. 2018.

[37] M Medvedeva et al. "Automatic judgement forecasting for pending applications of the European Court of Human Rights". In: *Proceedings of the fifth workshop on automated semantic analysis of*. 2021.

[38] Masha Medvedeva, Michel Vols, and Martijn Wieling. "Using machine learning to predict decisions of the European Court of Human Rights". en. In: *Artif. Intell. Law* 28.2 (June 2020), pp. 237–266.

[39] Masha Medvedeva, Martijn Wieling, and Michel Vols. "Rethinking the field of automatic prediction of court decisions". In: *Artificial Intelligence and Law* (Jan. 2022). DOI: 10.1007/s10506-021-09306-3.

[40] Masha Medvedeva, Martijn Wieling, and Michel Vols. "The Danger of Reverse-Engineering of Automated Judicial Decision-Making Systems". In: *CoRR* abs/2012.10301 (2020). arXiv: 2012.10301. URL: https://arxiv.org/abs/2012.10301.

[41] I W O N A Osmanska-Lipka. *Elements of Gestalt psychology in American cognitive linguistics*. 2013.

[42] Ugo Pagallo. "Three roads to complexity, AI and the law of robots: On crimes, contracts, and torts". In: *Lecture Notes in Computer Science*. Lecture notes in computer science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 48–60.

[43] E Pauwels and S W Denton. "The internet of bodies: life and death in the age of AI". In: *Cal. WL Rev* 55 (2018).

[44] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[45] Ayse Pinar Saygin, Ilyas Cicekli, and Varol Akman. "Turing test: 50 years later". In: *Minds and machines* 10.4 (2000), pp. 463–518.

[46] Tomas Pranckevičius and Virginijus Marcinkevičius. "Comparison of naive Bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification". In: *Balt. J. Mod. Comput.* 5.2 (2017).

[47] Peng Qi et al. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". In: *CoRR* abs/2003.07082 (2020). arXiv: 2003.07082. URL: https://arxiv.org/abs/2003.07082.

[48] Inioluwa Deborah Raji et al. "Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. FAT* '20. Barcelona, Spain: Association for Computing Machinery, 2020, pp. 33–44. ISBN: 9781450369367. DOI: 10.1145/3351095.3372873. URL: https://doi.org/10.1145/3351095.3372873.

[49] Marco Tulio Ribeiro et al. "Beyond Accuracy: Behavioral Testing of NLP models with CheckList". In: *CoRR* abs/2005.04118 (2020). arXiv: 2005.04118. URL: https://arxiv.org/abs/2005.04118.

[50] Ludovic Righetti, Raj Madhavan, and Raja Chatila. "Unintended consequences of biased robotic and artificial intelligence systems [ethical, legal, and societal issues]". In: *IEEE Robot. Autom. Mag.* 26.3 (Sept. 2019), pp. 11–13.

[51] Ludovic Righetti, Raj Madhavan, and Raja Chatila. "Unintended consequences of biased robotic and artificial intelligence systems [ethical, legal, and societal issues]". In: *IEEE Robotics & Automation Magazine* 26.3 (2019), pp. 11–13.

[52] Eugen Ruppert et al. "LawStats – Large-Scale German Court Decision Evaluation Using Web Service Classifiers". In: *Machine Learning and Knowledge Extraction*. Ed. by Andreas Holzinger et al. Cham: Springer International Publishing, 2018, pp. 212–222. ISBN: 978-3-319-99740-7.

[53] Boaz Shmueli et al. "Beyond Fair Pay: Ethical Implications of NLP Crowdsourcing". In: *CoRR* abs/2104.10097 (2021). arXiv: 2104.10097. URL: https://arxiv.org/abs/2104.10097.

[54] Tania Sourdin. In: *The University of New South Wales Law Journal* 41.4 (2018), pp. 1114–1133. ISSN: 0313-0096. URL: https://search.informit.org/doi/10.3316/informit.040979608613368.

[55] Robert J Spitzer. "Why is human self-consciousness different from artificial intelligence and animal consciousness?" en. In: *Ultim. Real. Mean.* 33.1-2 (Mar. 2010), pp. 5–27.

[56] Octavia-Maria Sulea et al. "Predicting the Law Area and Decisions of French Supreme Court Cases". In: *CoRR* abs/1708.01681 (2017). arXiv: 1708.01681. URL: http://arxiv.org/abs/1708.01681.

[57] Kevin Tobia, Aileen Nielsen, and Alexander Stremitzer. "When Does Physician Use of AI Increase Liability?" In: *Journal of Nuclear Medicine* 62.1 (2021), pp. 17–21. ISSN: 0161-5505. DOI: 10.2967/jnumed.120.256032. eprint: https://jnm.snmjournals.org/content/62/1/17.full.pdf. URL: https://jnm.snmjournals.org/content/62/1/17.

[58] *Transforming the Criminal Justice System: Court Delays*. YouTube, 2017. URL: https://www.youtube.com/watch?v=lm7ELdQ1_qc.

[59] Michael Benedict L. Virtucio et al. "Predicting Decisions of the Philippine Supreme Court Using Natural Language Processing and Machine Learning". In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 02. 2018, pp. 130–135. DOI: 10.1109/COMPSAC.2018.10348.

[60] Eugene Volokh. "Chief justice robots". In: *Duke LJ* 68 (2018), p. 1135.

[61] Alex Wang et al. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *CoRR*

abs/1804.07461 (2018). arXiv: 1804.07461. URL: http://arxiv.org/abs/1804.07461.

[62] C Williams. “Legal English and plain language: An introduction”. In: *ESP across Cultures* 1.1 (2004), pp. 111–124.

[63] Alice Woolley. “The resignation of Robin Camp: background and reflections from Canada”. In: *Legal Ethics* 20.1 (2017), pp. 134–137.

[64] Dean Young. “Your ways or our ways?: Addressing Canadian Neo-colonialism and restorative justice”. In: *Salus Journal* 7.2 (2019), pp. 85–108.

Appendices

Appendix A

Redacted Emails

Redacted email giving permission to download the cases from the SCC

[redacted]@scc-csc.ca>
to me, [redacted]

Mon, Aug 24, 2020, 2:33 PM

Hello Mr. Thomas Malley,

Your request for permission to download the database for the sole purpose of education and research falls under the *Reproduction of Federal Law Order*. However, at this time, the Court nor Lexum has the capacity to provide bulk access to the judgments. SCC Judgments may be downloaded manually from the site, in small batches, or using web crawlers programmed at a slow rate that does not exceed Lexum's bandwidth capacity.

A status update upon completion of your research project would be much appreciated.

Please note that this permission does not extend to commercial uses and further approval would be required from the Office of the Registrar of the Supreme Court of Canada (ORSCC) should your use of the data change.

Regards,

Alicia

[redacted]
Director | Directrice
Library Branch | Direction générale de la bibliothèque
Supreme Court of Canada | Cour suprême du Canada
301 Wellington Street | 301, rue Wellington
Ottawa, Ontario K1A 0J1
[redacted]@SCC-CSC.CA
[redacted]

Supreme Court of Canada | Cour suprême du Canada

Figure A.1: Permission to download the cases from the SCC website

[redacted]@lexum.com>
to me

Wed, Aug 26, 2020, 11:55 AM

Hi Thomas,

Crawling will not work as you will likely be banned by our traffic monitoring software. I will contact the SCC and they should get back to you on this.

[redacted]
Business Development / Développement des affaires
[redacted]@lexum.com

Figure A.2: Instructions against automated download

Appendix B

Full Results

Table B.1: Complete Set of Results

Run	Algorithm	Test_Size	NL	NH	MaxDF	Time to Prep	Time to Fit	Accuracy	F1
1	NB	0.1	1	1	0.3	3.5659	3.6228	38.6364	47.0588
2	SVM	0.1	1	1	0.3	3.4794	4.0630	36.3636	39.1304
3	KNN	0.1	1	1	0.3	3.5568	3.5903	36.3636	36.3636
4	LR	0.1	1	1	0.3	3.5659	3.7445	31.8182	40.0000
5	NB	0.1	1	1	0.5	3.6143	3.6661	31.8182	46.4286
6	SVM	0.1	1	1	0.5	3.6273	4.4118	4.4118	47.8261
7	KNN	0.1	1	1	0.5	3.5583	3.5960	36.3636	39.1304
8	LR	0.1	1	1	0.5	3.5502	3.7344	40.9091	48.0000
9	NB	0.1	1	1	0.7	3.6477	3.7046	31.8182	46.4286
10	SVM	0.1	1	1	0.7	3.7503	4.7135	38.6364	40.0000
11	KNN	0.1	1	1	0.7	3.6798	3.7227	47.7273	46.5116
12	LR	0.1	1	1	0.7	3.7228	3.9153	38.6364	44.8980
13	NB	0.2	1	1	0.3	3.7087	3.7538	40.9091	51.8519
14	SVM	0.2	1	1	0.3	3.5609	4.0102	40.9091	46.9388
15	KNN	0.2	1	1	0.3	3.5560	3.5855	35.2273	37.3626
16	LR	0.2	1	1	0.3	3.7214	3.8775	39.7727	49.5238
17	NB	0.2	1	1	0.5	3.6275	3.6734	42.0455	53.2110
18	SVM	0.2	1	1	0.5	3.6062	4.2136	48.8636	52.6316
19	KNN	0.2	1	1	0.5	3.5971	3.6319	44.3182	46.1538
20	LR	0.2	1	1	0.5	3.7348	3.8859	44.3182	52.4272
21	NB	0.2	1	1	0.7	3.6192	3.6649	42.0455	54.8673
22	SVM	0.2	1	1	0.7	3.6786	4.4312	47.7273	51.0638
23	KNN	0.2	1	1	0.7	3.6659	3.6999	44.3182	46.1538
24	LR	0.2	1	1	0.7	3.7238	3.8934	45.4545	53.8462
25	NB	0.3	1	1	0.3	3.4982	3.5341	45.8015	55.3459
26	SVM	0.3	1	1	0.3	3.5985	3.9426	45.0382	49.2958
27	KNN	0.3	1	1	0.3	3.7551	3.7840	40.4580	44.2857
28	LR	0.3	1	1	0.3	4.0848	4.2191	45.8015	54.1935
29	NB	0.3	1	1	0.5	3.6037	3.6446	47.3282	57.1429
30	SVM	0.3	1	1	0.5	3.7064	4.1719	45.8015	48.1752
31	KNN	0.3	1	1	0.5	3.5924	3.6173	47.3282	48.8889
32	LR	0.3	1	1	0.5	3.5548	3.6841	46.5649	55.6962
33	NB	0.3	1	1	0.7	3.5571	3.5940	48.8550	59.8802
34	SVM	0.3	1	1	0.7	3.5223	4.0901	48.0916	50.7246
35	KNN	0.3	1	1	0.7	3.6290	3.6549	46.5649	52.0548
36	LR	0.3	1	1	0.7	3.6536	3.7839	50.3817	58.0645
37	NB	0.4	1	1	0.3	3.6038	3.6323	47.4286	59.6491
38	SVM	0.4	1	1	0.3	3.5631	3.8146	45.7143	52.2613
39	KNN	0.4	1	1	0.3	3.5255	3.5445	43.4286	48.1675
40	LR	0.4	1	1	0.3	3.6638	3.7791	46.8571	57.9186
41	NB	0.4	1	1	0.5	3.6758	3.7067	48.0000	61.2766
42	SVM	0.4	1	1	0.5	3.6326	3.9692	43.4286	50.7463
43	KNN	0.4	1	1	0.5	3.7101	3.7300	45.1429	51.0204
44	LR	0.4	1	1	0.5	3.6757	3.7957	45.7143	57.7778
45	NB	0.4	1	1	0.7	3.6455	3.6754	45.7143	60.9053
46	SVM	0.4	1	1	0.7	3.5639	3.9765	44.0000	50.0000
47	KNN	0.4	1	1	0.7	3.5774	3.5983	48.5714	55.0000
48	LR	0.4	1	1	0.7	3.5714	3.6890	46.8571	58.2960
49	NB	0.1	2	2	0.3	12.2946	13.7696	36.3636	44.0000
50	SVM	0.1	2	2	0.3	11.8037	15.5769	47.7273	46.5116

Run	Algorithm	Test_Size	NL	NH	MaxDF	Time to Prep	Time to Fit	Accuracy	F1
51	KNN	0.1	2	2	0.3	11.8018	12.7249	47.7273	43.9024
52	LR	0.1	2	2	0.3	11.8028	18.9515	38.6364	44.8980
53	NB	0.1	2	2	0.5	11.7885	13.2476	36.3636	44.0000
54	SVM	0.1	2	2	0.5	12.0578	15.9365	45.4545	45.4545
55	KNN	0.1	2	2	0.5	12.5726	13.5449	38.6364	40.0000
56	LR	0.1	2	2	0.5	12.5747	19.7016	34.0909	43.1373
57	NB	0.1	2	2	0.7	11.9557	13.4048	36.3636	44.0000
58	SVM	0.1	2	2	0.7	11.9418	15.8075	43.1818	41.8605
59	KNN	0.1	2	2	0.7	12.3232	13.2673	50.0000	45.0000
60	LR	0.1	2	2	0.7	12.6323	18.6988	36.3636	44.0000
61	NB	0.2	2	2	0.3	11.7544	12.9104	40.9091	52.7273
62	SVM	0.2	2	2	0.3	11.9227	14.8421	45.4545	52.0000
63	KNN	0.2	2	2	0.3	12.2973	13.0565	44.3182	47.3118
64	LR	0.2	2	2	0.3	12.1286	18.0901	39.7727	53.0973
65	NB	0.2	2	2	0.5	11.4125	12.5553	42.0455	54.0541
66	SVM	0.2	2	2	0.5	11.5698	14.5007	46.5909	52.5253
67	KNN	0.2	2	2	0.5	12.1831	12.9261	44.3182	49.4845
68	LR	0.2	2	2	0.5	12.2354	18.0968	40.9091	53.5714
69	NB	0.2	2	2	0.7	11.4064	12.5510	42.0455	54.0541
70	SVM	0.2	2	2	0.7	12.1036	15.0673	47.7273	53.0612
71	KNN	0.2	2	2	0.7	12.3414	13.1049	48.8636	49.4382
72	LR	0.2	2	2	0.7	12.2502	17.2789	40.9091	53.5714
73	NB	0.3	2	2	0.3	11.4482	12.3689	41.9847	55.8140
74	SVM	0.3	2	2	0.3	11.8992	14.1702	47.3282	54.9020
75	KNN	0.3	2	2	0.3	11.6957	12.2681	45.8015	47.4074
76	LR	0.3	2	2	0.3	11.6599	16.2595	42.7481	57.6271
77	NB	0.3	2	2	0.5	11.2042	12.1589	41.2214	55.4913
78	SVM	0.3	2	2	0.5	12.0166	14.3012	47.3282	54.9020
79	KNN	0.3	2	2	0.5	11.9721	12.5687	48.0916	52.1127
80	LR	0.3	2	2	0.5	11.7134	16.3295	43.5115	58.4270
81	NB	0.3	2	2	0.7	11.2256	12.1404	41.2214	55.4913
82	SVM	0.3	2	2	0.7	12.0438	14.3695	47.3282	54.3046
83	KNN	0.3	2	2	0.7	12.3056	12.8951	47.3282	48.1203
84	LR	0.3	2	2	0.7	12.1240	16.1461	45.8015	58.9595
85	NB	0.4	2	2	0.3	11.0381	11.7511	46.2857	61.4754
86	SVM	0.4	2	2	0.3	11.7944	13.4219	44.5714	56.1086
87	KNN	0.4	2	2	0.3	11.7304	12.1774	47.4286	51.5789
88	LR	0.4	2	2	0.3	11.7916	15.7191	45.1429	61.2903
89	NB	0.4	2	2	0.5	10.9741	11.6584	46.2857	61.4754
90	SVM	0.4	2	2	0.5	11.7092	13.3725	44.5714	56.1086
91	KNN	0.4	2	2	0.5	11.8137	12.2525	46.2857	53.0000
92	LR	0.4	2	2	0.5	11.4040	15.3275	45.1429	61.2903
93	NB	0.4	2	2	0.7	11.0470	11.7272	46.2857	61.4754
94	SVM	0.4	2	2	0.7	11.7548	13.4047	45.1429	55.5556
95	KNN	0.4	2	2	0.7	11.7219	12.1627	48.5714	50.0000
96	LR	0.4	2	2	0.7	11.8127	15.0745	44.0000	60.4839
97	NB	0.1	3	3	0.3	17.7444	20.9197	34.0909	43.1373
98	SVM	0.1	3	3	0.3	18.9373	23.5448	47.7273	41.0256
99	KNN	0.1	3	3	0.3	18.5589	20.4530	43.1818	39.0244

Run	Algorithm	Test_Size	NL	NH	MaxDF	Time to Prep	Time to Fit	Accuracy	F1
100	LR	0.1	3	3	0.3	18.6303	30.5040	36.3636	48.1481
101	NB	0.1	3	3	0.5	17.6932	20.6720	34.0909	43.1373
102	SVM	0.1	3	3	0.5	18.6888	23.2680	47.7273	41.0256
103	KNN	0.1	3	3	0.5	18.1571	20.0413	43.1818	41.8605
104	LR	0.1	3	3	0.5	18.9070	30.9864	34.0909	45.2830
105	NB	0.1	3	3	0.7	17.8603	21.1039	34.0909	43.1373
106	SVM	0.1	3	3	0.7	19.7102	24.3418	47.7273	41.0256
107	KNN	0.1	3	3	0.7	18.9957	20.9148	40.9091	40.9091
108	LR	0.1	3	3	0.7	18.4493	30.6827	34.0909	45.2830
109	NB	0.2	3	3	0.3	16.9561	19.2254	36.3636	50.8772
110	SVM	0.2	3	3	0.3	17.8074	21.3147	52.2727	53.3333
111	KNN	0.2	3	3	0.3	17.7513	19.2088	53.4091	53.9326
112	LR	0.2	3	3	0.3	17.8494	27.2853	39.7727	55.4622
113	NB	0.2	3	3	0.5	16.3005	18.4991	38.6364	53.4483
114	SVM	0.2	3	3	0.5	16.4132	19.8511	54.5455	54.5455
115	KNN	0.2	3	3	0.5	16.4244	17.8407	52.2727	53.3333
116	LR	0.2	3	3	0.5	16.3827	25.3271	40.9091	56.6667
117	NB	0.2	3	3	0.7	16.8407	19.0825	38.6364	53.4483
118	SVM	0.2	3	3	0.7	16.8843	20.4247	54.5455	54.5455
119	KNN	0.2	3	3	0.7	17.8488	19.3250	52.2727	53.3333
120	LR	0.2	3	3	0.7	17.6766	27.1291	40.9091	56.6667
121	NB	0.3	3	3	0.3	16.5396	18.3323	42.7481	57.6271
122	SVM	0.3	3	3	0.3	17.2423	19.9759	47.3282	54.9020
123	KNN	0.3	3	3	0.3	17.1864	18.3233	51.9084	55.3191
124	LR	0.3	3	3	0.3	17.2751	25.1684	42.7481	59.4595
125	NB	0.3	3	3	0.5	16.4701	18.2560	42.7481	57.6271
126	SVM	0.3	3	3	0.5	16.9686	19.7141	50.3817	56.3758
127	KNN	0.3	3	3	0.5	17.1868	18.3139	53.4351	56.7376
128	LR	0.3	3	3	0.5	17.1361	24.7694	41.9847	58.6957
129	NB	0.3	3	3	0.7	16.2721	18.0191	42.7481	57.6271
130	SVM	0.3	3	3	0.7	16.8715	19.6781	49.6183	56.0000
131	KNN	0.3	3	3	0.7	17.2299	18.3562	53.4351	56.7376
132	LR	0.3	3	3	0.7	17.4167	25.1639	41.9847	58.6957
133	NB	0.4	3	3	0.3	15.3830	16.6586	44.0000	60.1626
134	SVM	0.4	3	3	0.3	16.4088	18.3333	44.0000	57.3913
135	KNN	0.4	3	3	0.3	16.2382	17.0506	46.8571	55.0725
136	LR	0.4	3	3	0.3	16.5557	22.0889	43.4286	60.5578
137	NB	0.4	3	3	0.5	15.4639	16.8003	44.0000	60.1626
138	SVM	0.4	3	3	0.5	16.6533	18.6094	45.1429	57.8947
139	KNN	0.4	3	3	0.5	16.3884	17.2068	48.5714	56.7308
140	LR	0.4	3	3	0.5	16.4695	21.9992	43.4286	60.5578
141	NB	0.4	3	3	0.7	15.4533	16.7280	44.0000	60.1626
142	SVM	0.4	3	3	0.7	16.5123	18.5070	45.1429	57.8947
143	KNN	0.4	3	3	0.7	16.4028	17.2049	48.5714	56.7308
144	LR	0.4	3	3	0.7	16.4970	21.8381	43.4286	60.5578

Appendix C

Case Exclusions

2019 - 27 Cases Excluded

Excluded because there were more than one decision

- Denis v. CÃ’tÃ©, 2019 SCC 44
- MontrÃ©al (Ville) v. Octane StratÃ©gie inc., 2019 SCC 57
- R. v. Stillman, 2019 SCC 40

Excluded for lack of data/information, <than 10 pages

- R. v. Demedeiros, 2019 SCC 11, [2019] 1 S.C.R. 568
- R. v. George-Nurse, 2019 SCC 12, [2019] 1 S.C.R. 570
- Barer v. Knight Brothers LLC, 2019 SCC 13, [2019] 1 S.C.R. 573
- R. v. Snelgrove, 2019 SCC 16, [2019] 2 S.C.R. 98
- R. v. Kelsie, 2019 SCC 17, [2019] 2 S.C.R. 101

- TELUS Communications Inc. v. Wellman, 2019 SCC 19, [2019] 2 S.C.R. 144
- R. v. Beaudry, 2019 SCC 2, [2019] 1 S.C.R. 95
- R. v. Thanabalasingham, 2019 SCC 21, [2019] 2 R.C.S. 317
- R. v. Dâ€™Amico, 2019 SCC 23, [2019] 2 S.C.R. 394
- R. v. J.M., 2019 SCC 24, [2019] 2 S.C.R. 396
- R. v. Larue, 2019 SCC 25, [2019] 2 S.C.R. 398
- R. v. Wakefield, 2019 SCC 26, [2019] 2 S.C.R. 400
- R. v. W.L.S., 2019 SCC 27, [2019] 2 S.C.R. 403
- R. v. Fedyck, 2019 SCC 3, [2019] 1 S.C.R. 97
- Christine DeJong Medicine Professional Corp. v. DBDC Spadina Ltd., 2019 SCC 30, [2019] 2 S.C.R. 530
- R. v. Omar, 2019 SCC 32, [2019] 2 S.C.R. 576
- R. v. C.J., 2019 SCC 8, [2019] 1 S.C.R. 484
- R. v. Blanchard, 2019 SCC 9, [2019] 1 S.C.R. 486
- R. v. Collin, 2019 SCC 64
- G-en.docx
- R. v. James, 2019 SCC 52
- R. v. Kernaz, 2019 SCC 48
- R. v. M.R.H., 2019 SCC 46
- R. v. Shlah, 2019 SCC 56

2018 - 21 Cases Excluded

Excluded because there were more than one decision

- Reference re PanCanadian Securities Regulation, 2018 SCC 48, [2018] 3 S.C.R. 189

Excluded for lack of data/information, <than 10 pages

- R. v. Seipp, 2018 SCC 1, [2018] 1 S.C.R. 3
- R. v. Black, 2018 SCC 10, [2018] 1 S.C.R. 265
- International Brotherhood of Electrical Workers (IBEW) Local 773 v. Lawrence, 2018 SCC 11, [2018] 1 S.C.R. 267
- R. v. R.A., 2018 SCC 13, [2018] 1 S.C.R. 307
- R. v. Cain, 2018 SCC 20, [2018] 1 S.C.R. 631
- R. v. Stephan, 2018 SCC 21, [2018] 1 S.C.R. 633
- R. v. Colling, 2018 SCC 23, [2018] 1 S.C.R. 692
- R. v. Gulliver, 2018 SCC 24, [2018] 1 S.C.R. 694
- R. v. Gagnon, 2018 SCC 41, [2018] 3 S.C.R. 3
- R. v. Normore, 2018 SCC 42, [2018] 3 S.C.R. 5
- R. v. Awashish, 2018 SCC 45, [2018] 3 S.C.R. 87
- Callidus Capital Corp. v. Canada, 2018 SCC 47, [2018] 3 S.C.R. 186
- R. v. Youssef, 2018 SCC 49, [2018] 3 S.C.R. 259

- R. v. Ajise, 2018 SCC 51, [2018] 3 S.C.R. 301
- R. v. Culotta, 2018 SCCÂ 57, [2018] 3 S.C.R. 597
- R. v. Quartey, 2018 SCCÂ 59, [2018] 3 S.C.R. 687
- R. v. A.R.J.D., 2018 SCC 6, [2018] 1 S.C.R. 218
- R. v. G.T.D., 2018 SCC 7, [2018] 1 S.C.R. 220
- R. v. A.G.W., 2018 SCC 9, [2018] 1 S.C.R. 263
- Cyr-Langlois bil.docx

2017 - 17 Cases Excluded

Excluded for lack of data/information, <than 10 pages

- R. v. Brown, 2017 SCCÂ 10, [2017] 1 S.C.R. 166
- R. v. Olotu, 2017 SCCÂ 11, [2017] 1 S.C.R. 168
- R. v. Peers, 2017 SCC 13, [2017] 1 S.C.R. 196
- R. v. Aitkens, 2017 SCC 14, [2017] 1 S.C.R. 199
- R. v. S.B., 2017 SCCÂ 16, [2017] 1 S.C.R. 248
- R. v. Savard, 2017 SCC 21, [2017] 1 S.C.R. 400
- Pintea v. Johns, 2017 SCC 23, [2017] 1 S.C.R. 470
- Lajeunesse (Re), 2017 SCCÂ 24, [2017] 1 S.C.R. 473
- R. v. Hunt, 2017 SCC 25, [2017] 1 S.C.R. 476
- R. v. Clark, 2017 SCC 3, [2017] 1 S.C.R. 86

- R. v. BÃ©dard, 2017 SCC 4, [2017] 1 S.C.R. 89
- R. v. Bourgeois, 2017 SCCÂ 49, [2017] 2 S.C.R. 287
- R. v. Natewayes, 2017 SCC 5, [2017] 1 S.C.R. 91
- R. v. Robinson, 2017 SCCÂ 52, [2017] 2 S.C.R. 382
- R. v. Millington, 2017 SCCÂ 53, [2017] 2 S.C.R. 384
- R. v. Clifford, 2017 SCC 9, [2017] 1 S.C.R. 164
- George en.docx

2016 - 20 Cases Excluded

Excluded because there were more than one decisions

- Daniels v. Canada (Indian Affairs and Northern Development), 2016 SCC 12, [2016] 1 S.C.R. 99
- Endean v. British Columbia, 2016 SCCÂ 42, [2016] 2 S.C.R. 162
- British Columbia (Workersâ€™ Compensation Appeal Tribunal) v. Fraser Health Authority, 2016 SCC 25, [2016] 1 S.C.R. 587

Excluded because they were motions

- Carter v. Canada (Attorney General), 2016 SCC 4, [2016] 1 S.C.R. 13
- Brine v. Industrial Alliance Insurance and Financial Services Inc., 2016 SCC 9, [2016] 1 S.C.R. 72

Excluded for lack of data/information, <than 10 pages

- British Columbia Teachersâ€™ Federation v. British Columbia, 2016 SCC 49, [2016] 2 S.C.R. 407
- R. v. C.K-D., 2016 SCC 41, [2016] 2 S.C.R. 160
- Canadian Pacific Railway Co. v. Canada (Attorney General), 2016 SCC 1, [2016] 1 S.C.R. 6
- R. v. Diamond, 2016 SCCÂ 46, [2016] 2 S.C.R. 291
- R. v. Gagnon, 2016 SCC 6, [2016] 1 S.C.R. 25
- R. v. Knapczyk, 2016 SCCÂ 10, [2016] 1 S.C.R. 78
- R. v. LalibertÃ©, 2016 SCCÂ 17, [2016] 1 S.C.R. 270
- R. v. Meer, 2016 SCC 5, [2016] 1 S.C.R. 23
- R. v. Newman, 2016 SCC 7, [2016] 1 S.C.R. 27
- R. v. Rowson, 2016 SCC 40, [2016] 2 S.C.R. 158
- R. v. Seruhungo, 2016 SCC 2, [2016] 1 S.C.R. 9
- R. v. Shaoulle, 2016 SCCÂ 16, [2016] 1 S.C.R. 268
- R. v. Spicer, 2016 SCC 3, [2016] 1 S.C.R. 11
- Urban Communications Inc. v. BCNET Networking Society, 2016 SCC 45, [2016] 2 S.C.R. 289
- R. v. Vassell, 2016 SCC 26, [2016] 1 S.C.R. 625

2015 - 14 Cases Excluded

Excluded because there were more than one decisions

- Canadian Imperial Bank of Commerce v. Green, 2015 SCC 60, [2015] 3 S.C.R. 801

Excluded because there was no conclusion

- Saskatchewan (Attorney General) v. Lemare Lake Logging Ltd., 2015 SCC 53, [2015] 3 S.C.R. 419

Excluded for lack of data/information, <than 10 pages

- R. v. Goleski, 2015 SCC 6, [2015] 1 S.C.R. 399
- R. v. Hecimovic, 2015 SCC 54, [2015] 3 S.C.R. 483
- Bowden Institution v. Khadr, 2015 SCCÂ 26, [2015] 2 S.C.R. 325
- R. v. M.J.B., 2015 SCC 48, [2015] 3 S.C.R. 321
- R. v. McKenna, 2015 SCCÂ 63, [2015] 3 S.C.R. 1087
- R. v. Riar, 2015 SCC 50, [2015] 3 S.C.R. 325
- S.H. v. Quebec (Emploi et SolidaritÃ© sociale), 2015 SCC 66, [2016] 1 S.C.R. 3
- R. v. Sanghera, 2015 SCC 13, [2015] 1 S.C.R. 691
- Sanofi-Aventis v. Apotex Inc., 2015 SCCÂ 20, [2015] 2 S.C.R. 136
- Zurich Insurance Co. v. Chubb Insurance Co. of Canada, 2015 SCCÂ 19, [2015] 2 S.C.R. 134

Excluded because they were duplicates, same docket numbers and missing information

- Barnaby en(1).docx

- Caplin en(1).docx
- Riesberry en(1).docx

2014 - 26 Cases Excluded

Excluded because there were more than one decisions

- R. v. MacDonald, 2014 SCC 3, [2014] 1 S.C.R. 37
- Trial Lawyers Association of British Columbia v. British Columbia (Attorney General), 2014 SCC 59, [2014] 3 S.C.R. 31

Excluded because there was no clear decision

- Reference re Supreme Court Act, ss. 5 and 6, 2014 SCCÂ 21, [2014] 1 S.C.R. 433
- Reference re Senate Reform, 2014 SCC 32

Excluded because they were motions

- Stubicar v. Canada (Public Safety and Emergency Preparedness), 2014 SCCÂ 38, [2014] 2 S.C.R. 104

Excluded for lack of data/information, <than 10 pages

- R. v. Auclair, 2014 SCC 6, [2014] 1 S.C.R. 83
- British Columbia Teachersâ€™ Federation v. British Columbia Public School Employersâ€™ Association, 2014 SCC 70, [2014] 3 S.C.R. 492
- R. v. Bouchard, 2014 SCC 64, [2014] 3 S.C.R. 283
- R. v. Day, 2014 SCC 74, [2014] 3 S.C.R. 614

- Dionne v. Commission scolaire des Patriotes, 2014 SCC 33, [2014] 1 S.C.R. 765
- R. v. Dunn, 2014 SCC 69, [2014] 3 S.C.R. 490
- R. v. Flaviano, 2014 SCC 14, [2014] 1 S.C.R. 270
- R. v. Hogg, 2014 SCC 18, [2014] 1 S.C.R. 344
- R. v. Jackson, 2014 SCC 30, [2014] 1 S.C.R. 672
- R. v. James, 2014 SCC 5, [2014] 1 S.C.R. 80
- R. v. Koczab, 2014 SCC 9, [2014] 1 S.C.R. 138
- R. v. Leinen, 2014 SCC 23, [2014] 1 S.C.R. 500
- R. v. LÃ©pine, 2014 SCC 65, [2014] 3 S.C.R. 285
- R. v. MacLeod, 2014 SCC 76, [2014] 3 S.C.R. 619
- R. v. Mohamed, 2014 SCC 63, [2014] 3 S.C.R. 280
- R. v. Vokurka, 2014 SCC 22, [2014] 1 S.C.R. 498
- R. v. Waite, 2014 SCC 17, [2014] 1 S.C.R. 341
- R. v. W.E.B., 2014 SCC 2, [2014] 1 S.C.R. 34
- R. v. Wilcox, 2014 SCC 75, [2014] 3 S.C.R. 616
- R. v. Wills, 2014 SCC 73, [2014] 3 S.C.R. 612
- R. v. Yelle, 2014 SCC 10, [2014] 1 S.C.R. 140

2013 - 17 Cases Excluded

Excluded for lack of data/information, <than 10 pages

- R. v. Manning, 2013 SCC 1, [2013] 1 S.C.R. 3
- R. v. Taylor, 2013 SCC 10, [2013] 1 S.C.R. 465
- R. v. Mailhot, 2013 SCC 17, [2013] 2 S.C.R. 96
- R. v. LÃ©vesque, 2013 SCC 20, [2013] 2 S.C.R. 176
- R. v. Murphy, 2013 SCC 21, [2013] 2 S.C.R. 178
- R. v. MacIntosh, 2013 SCC 23, [2013] 2 S.C.R. 200
- R. v. G.M., 2013 SCC 24, [2013] 2 S.C.R. 202
- R. v. Ibanescu, 2013 SCC 31, [2013] 2 S.C.R. 400
- Quebec (Attorney General) v. A, 2013 SCC 5, [2013] 1 S.C.R. 61
- R. v. R.L., 2013 SCC 54, [2013] 3 S.C.R. 418
- R. v. BÃ©langer, 2013 SCC 7, [2013] 1 S.C.R. 401
- R. v. Blacklaws, 2013 SCC 8, [2013] 1 S.C.R. 403

Excluded because there were more than one decisions

- Sun-Rype Products Ltd. v. Archer Daniels Midland Company, 2013 SCC 58, [2013] 3 S.C.R. 545
- Sun Indalex Finance, LLC v. United Steelworkers, 2013 SCC 6, [2013] 1 S.C.R. 271
- Wood v. Schaeffer, 2013 SCC 71, [2013] 3 S.C.R. 1053
- Canada (Attorney General) v. Bedford, 2013 SCC 72, [2013] 3 S.C.R. 1101
- Cinar Corporation v. Robinson, 2013 SCC 73, [2013] 3 S.C.R. 1168

2012 - 14 Cases Excluded

Excluded because there were more than one decisions

- R. v. Ipeelee, 2012 SCC 13, [2012] 1 S.C.R. 433
- Annapolis County District School Board v. Marshall, 2012 SCC 27, [2012] 2 S.C.R. 84
- Opitz v. Wrzesnewskyj, 2012 SCC 55, [2012] 3 S.C.R. 76

Excluded for lack of data/information, <than 10 pages

- R. v. Rochon, 2012 SCC 50, [2012] 2 S.C.R. 673
- R. v. P.D.T., 2012 SCC 62, [2012] 3 S.C.R. 394
- R. c. D.J.W., 2012 SCC 63, [2012] 3 S.C.R. 396
- Construction Labour Relations v. Driver Iron Inc., 2012 SCCÂ 65, [2012] 3 S.C.R. 405
- Momentous.ca Corp. v. Canadian American Association of Professional Baseball Ltd., 2012 SCC 9, [2012] 1 S.C.R. 359
- 2012SCC1.docx
- 2012SCC11.docx
- 2012SCC15.docx

Excluded because they were motions

- 2012SCC25.docx

2011 - 15 Cases Excluded

Excluded for lack of data/information, <than 10 pages

- 2011SCC14.docx
- R. v. St-Onge, 2011 SCC 16, [2011] 1 S.C.R. 625
- R. v. Reynolds, 2011 SCC 19, [2011] 1 S.C.R. 693
- R. v. Loewen, 2011 SCC 21, [2011] 2 S.C.R. 167
- R. v. V.Y., 2011 SCC 22, [2011] 2 S.C.R. 173
- R. v. E.M.W., 2011 SCC 31, [2011] 2 S.C.R. 542
- 2011SCC41.docx
- 2011SCC49.docx
- 2011SCC50.docx
- 2011SCC55.docx
- 2011SCC57.docx
- R. v. Bruce, 2011 SCC 4

Excluded because they were motions

- 2011SCC33.docx

Excluded because there were more than one decisions

- R. v. Imperial Tobacco Canada Ltd., 2011 SCC 42, [2011] 3 S.C.R. 45

Excluded because no clear decision

- Reference re Securities Act, 2011 SCC 66, [2011] 3 S.C.R. 837