Hastings Science and Technology Law Journal

Volume 11
Number 1 *Winter 2020*

Article 3

Winter 2020

Using Machine Learning on Legal Matters: Paying Attention to the Data Behind the Curtain

Robert Keeling

Rishi Chhatwal

Nathaniel Huber-Fliflet

Jianping Zhang

Haozhen Zhao

Recommended Citation

Robert Keeling, Rishi Chhatwal, Nathaniel Huber-Fliflet, Jianping Zhang, and Haozhen Zhao, *Using Machine Learning on Legal Matters: Paying Attention to the Data Behind the Curtain*, 11 HASTINGS SCI. & TECH. L.J. 9 (2020).
Available at: https://repository.uchastings.edu/hastings_science_technology_law_journal/vol11/iss1/3

Using Machine Learning on Legal Matters: Paying Attention to the Data Behind the Curtain

ROBERT KEELING, RISHI CHHATWAL, NATHANIEL HUBER-FLIFLET, JIANPING ZHANG, AND HAOZHEN ZHAO

The following article offers key insights, previously undisclosed to the legal community, on how to improve the burdensome document review process through the use of machine learning, also known as predictive coding or technology assisted review (TAR). Document review has become particularly challenging because the volume of electronically stored information has grown exponentially in the past decade. Although document review has become increasingly time-intensive and expensive, employing machine learning can ease the burden of document review for counsel and clients. Machine learning uses computer algorithms to identify potentially relevant documents during discovery. The goal of machine learning is to reduce the manual review by attorneys of irrelevant and non-responsive documents.

Understanding the technical aspects of machine learning is essential for efficient document review in modern litigation. The carefully constructed experiments presented in this article shed light on how best to design predictive models. The authors of this article performed nearly 34,000 experiments to determine the best overall combinations of algorithms and backend settings for predictive modeling effectiveness. These experiments used six data sets from real cases across a variety of industries. The results of these experiments demonstrate that the current use of machine learning in legal matters is inefficient. Significant improvements can be made to basic settings that have the potential to greatly improve the performance of the algorithms and save literally thousands of hours of attorney time that is currently spent needlessly reviewing irrelevant documents.

This article both introduces the basics of the machine learning process and delves into the details of its technical settings. First, this article outlines the machine learning process and introduces the different types of parameter settings and algorithms involved in the process. Second, the article describes the experiments and data sets. Finally, the article reports the results of the experiments and highlights the key parameters that have the most significant influence on improving the accuracy of machine learning models. Because this article covers both the foundation of machine learning and the insights from our experiments, the article is of interest to practitioners with a wide range of experiences (or lack thereof) with machine learning.

The abstract accompanying this submission provides more helpful summary information. We hope that you will accept this article for publication.

Abstract

Young lawyers and seasoned attorneys alike will face the challenges of information management in large-scale litigation. As more and more company data is stored electronically, e-discovery challenges grow more complex and expensive. Companies and counsel may turn to machine learning in order to expedite the daunting document review process. Machine learning uses high-speed supervised learning algorithms to categorize documents into groups such as relevant to litigation, unresponsive, and privileged in order to assist counsel with document review. Machine learning operates similarly to your email spam filter: the filter differentiates between relevant content and unwanted email creating two categories of email types. In order to maximize the power of machine learning, practitioners need to understand the underlying technical aspects of the process. This article explains the findings of 33,600 recently-conducted experiments that demonstrate an important and previously unknown insight in the sphere of machine learning: not all machine learning tools provide similar results. Varying combinations of backend settings will result in significantly different results and dramatically impact the accuracy and effectiveness of machine learning.

Although attorneys have begun to use machine learning for document review, many commercially available tools do not use the technical settings that deliver the best results. As a result, thousands of hours of attorney time is wasted reviewing documents that are not responsive to the discovery requests of that particular case. The findings described in this article challenge several misconceptions about how to best design predictive models. In particular, the results of the experiments suggest that the long-standing preference for one machine learning algorithm – Support Vector Machine – over another may be misguided. Additionally, simply using less not relevant documents to train the predictive model, can result in a substantially more effective model, given a typical legal case data set. The following article will discuss these insights,

among many others, to help practitioners understand how to improve the outcome of their machine learning process.

Introduction

Information management has become a significant challenge because the global volume of electronically stored information has grown exponentially since 2010.[1] The amount of data that large companies store electronically today can total hundreds of times more than every book in the Library of Congress.[2] In the litigation context, companies frequently spend millions of dollars identifying and producing responsive, electronically-stored documents during discovery.[3] As many associates at law firms know, the document review process is incredibly time-consuming. The document review process is also the largest expense associated with finding relevant information from a large volume of information.[4]

The costs involved in this manual review have grown dramatically as more information is stored electronically. Lawyers spend countless hours reviewing documents to respond to routine discovery requests. A 2013 study from Microsoft provides context for the sheer magnitude of documents requiring storage and review in preparation for trial. The Microsoft study revealed that Microsoft is forced to store an average of 60 million pages each time a party brings a case against the company.[5] As a case progresses, Microsoft estimated that it was permitted to narrow that 60 million figure down to about 350,000 pages after filtering by issue, source and dates.[6] The company hires teams of lawyers to manually review those documents, and the attorneys end up finding around 87,500 pages that are

[1] Steve Lohr, *The Age of Big Data*, N.Y. TIMES, Feb. 12, 2012, at SR1, available at: http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html).

[2] "Data, data everywhere," *The Economist*, 25 Feb. 2010, http://www.economist.com/node/15557443?story_id=15557443.

[3] In a survey to RAND, parties from 57 cases reported spending between $17,000 and $27 million to produce electronically stored information. Nicholas M. Pace & Laura Zakaras, *Where the Money Goes: Understanding Litigant Expenditures for Producing Electronic Discovery*, RAND at 17 (2012), http://www.rand.org/content/dam/rand/pubs/monographs/2012/RAND_MG1208.pdf

[4] Parties from 57 cases reported in a survey to RAND that the task of reviewing electronically stored information accounted for 73 percent of production costs. *Id.* at xv.

[5] Microsoft Corporate Blogs, *Needles in Haystacks: The Secret Burden Holding Back our Economy*, THE OFFICIAL MICROSOFT BLOG (Nov. 25, 2013), https://blogs.microsoft.com/on-the-issues/2013/11/25/needles-in-haystacks-the-secret-burden-holding-back-our-economy/#pwAdZg3Fr7Clxwi1.99.

[6] *Id.*

arguably relevant to the issues in the case and produced to the other side.[7] Of the 60 million pages that the company starts with, only 88 end up making it to court.[8] Microsoft estimates that it has spent around $600 million on outside services to help with discovery in the past decade or so, not including internal systems and employees dedicated to managing it.[9] Microsoft is not alone. Companies generally expend enormous resources in the maintenance and review of documents for discovery. A RAND Corporation study revealed that the document review process is responsible for the majority of e-discovery costs, with document review typically accounting for about 73% of all production costs, collection consisting of roughly 8% of expenditures, and costs for processing amounting to about 19% in typical cases.[10] These expenses represent a distinct and significant expense for companies today.

The burdens of document maintenance, production, and review even spurred a recent change in the 2015 amendments to the Federal Rules of Civil Procedure. The 2015 amendments added a requirement that discovery must be proportional to the needs of the case and consider the benefit versus burden of obtaining the information.[11] While proportionality has long been a consideration in the discovery rules, the determination of the burden on each party to produce certain information has evolved as more data has been stored electronically. The notes to the new amendments explain: "The burden or expense of proposed discovery should be determined in a realistic way. This includes the burden or expense of producing electronically stored information. Computer-based methods of searching such information continue to develop, particularly for cases involving large volumes of electronically stored information. Courts and parties should be willing to consider the opportunities for reducing the burden or expense of discovery as reliable means of searching electronically stored information become available."[12] The Rules both acknowledge the costly process of e-discovery and the technology that can assist with reducing the expense of e-discovery.

[7] *Id.*

[8] *Id.*

[9] *See Id.*

[10] Nicholas M. Pace & Laura Zakaras, *Where the Money Goes: Understanding Litigant Expenditures for Producing Electronic Discovery*, RAND at 17 (2012), http://www.rand.org/content/dam/rand/pubs/monographs/2012/RAND_MG1208.pdf.

[11] Fed. R. Civ. P. 26(b)(1).

[12] Fed. R. Civ. P. 26(b)(1) advisory committee's note to 2015 amendments.

To cull through massive volumes of data more efficiently, companies can turn to a process called machine learning, which uses supervised learning algorithms to categorize documents into classes—e.g., relevant to litigation, unresponsive, or privileged—based on similar document examples. More specifically, machine learning, also known as technology-assisted review or predictive coding, is a process by which "computers are programmed to search large quantities of documents . . . to mimic the document selection process of a knowledgeable, human document review."[13] This technology enables parties to conduct document review "faster and without many of the dangers of human error," and has been described as a "fundamental change in the way discovery is conducted."[14] While machine learning is already highly valued in some litigation settings, companies are now using machine learning in other legal and business matters—from responding to government inquiries to conducting due diligence in a merger. Though machine learning has become more common, the technical aspects of machine learning are not widely understood. In order to prepare for trial in a cost-effective manner, modern lawyers need to understand the technology available to assist with document review. This article seeks to open the black box that obscures the inner workings of the machine learning process. The studies explained in this article inform practitioners about how to improve the performance of the machine learning process as applied to their own document review challenges.

Although machine learning has grown more common in the litigation setting, few attorneys understand how to improve the results of their technology-assisted document reviews. Two variables in the machine learning process could shape the effectiveness of a predictive model: (1) the specific documents used to train the predictive model or (2) the backend technical settings used to develop the predictive model. In the field of technology-assisted review, the long-standing presumption is that the example documents used to train the machine learning algorithms can improve the accuracy and effectiveness of machine learning, while technical modifications cannot.[15] For example, scholars Maura Grossman

[13] Charles Yablon & Nick Landsman-Roos, *Predictive Coding: Emerging Questions and Concerns*, 64 S.C. L. REV. 633, 634 (2013).

[14] *Id.*

[15] *See* Maura Grossman & Gordan Cormack, *Technology-Assisted Review in E-Discovery Can Be More Effective and More Efficient Than Exhaustive Manual Review*, XVII RICH. J.L. & TECH. 11 (2011), http://jolt.richmond.edu/v17i3/article11.pdf (offering evidence that technology-assisted review yields superior results to manual review) *and* Maura Grossman & Gordan Cormack, *Evaluation of Machine-Learning Protocols for Technology-Assisted Review in Electronic Discovery*, plg2.cs.uwaterloo.ca/~gvcormac/

and Gordan Cormack have tested whether training documents should be selected at random or by using non-random methods in order to make machine learning more effective.[16] Grossman and Cormack have found that using non-random training methods to select the training documents, such as keyword searching is better at reducing required attorney review for passive learning, the same type of machine learning we implemented in these experiments.[17] The experiments by Grossman and Cormack that isolated the training document variable did not specifically test the technical settings involved in the machine learning process, however. The experiments described in this article, on the other hand, show that technical adjustments can have a dramatic impact on results.

Choosing ineffective technology settings for a machine-learning model can cause users to miss critical documents and increase costs by requiring expensive manual review of additional documents that are not relevant. In some cases, using ineffective technology settings will decrease the predictive model's precision[18] by more than 32%, which can reduce cost savings by more than 59%. In a litigation matter with one million documents—a common proposition for companies today—improving the efficiency of a model by even 5% can result in 50,000 fewer documents and 500 fewer hours for traditional attorney review.[19]

We performed nearly 34,000 experiments to determine which technical settings involved in the machine learning process could deliver the best results. Our experiments used various combinations of technology settings on three data sets from real legal matters to generate predictive models. This article (i) outlines the machine learning process and introduces different types of backend technical settings; (ii) describes the experiments and the data sets used; and (iii) reports our results and findings.

The insights contained within this article, if applied correctly, can significantly reduce costs, increase efficiency, and enhance privacy protections. Furthermore, our insights are broadly applicable to all three

calstudy/study/sigir2014-cormackgrossman.pdf (testing whether training documents should be selected at random or using non-random methods).

[16] *See* Maura Grossman & Gordan Cormack, *Evaluation of Machine-Learning Protocols for Technology-Assisted Review in Electronic Discovery*, plg2.cs.uwaterloo.ca /~gvcormac/calstudy/study/sigir2014-cormackgrossman.pdf.

[17] *See id.*

[18] Precision is the percentage of documents the predictive model marked as relevant that are relevant. It is a measurement to help determine which documents to review first, those that are strong QC candidates and how many irrelevant documents will require review to identify the relevant documents.

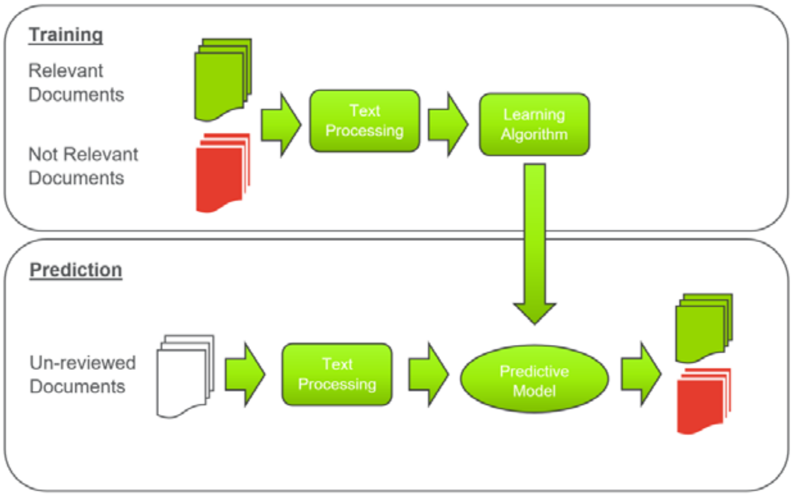
[19] *Id.* at 50.

major uses of predictive coding – identifying responsive and privileged documents during discovery, complying with production requests from government agencies, and conducting due diligence for corporate mergers.

Machine Learning Process

Machine learning in the document review context uses supervised learning algorithms to sift through a large document set and determine which documents are likely to be most relevant to the matter and which documents are not. As litigation matters have begun to involve more electronically stored data, the practice of using machine learning to assist with document review has grown more popular. The machine learning process basically uses a sample of documents to train a supervised learning algorithm with the information the algorithm needs to categorize the rest of the documents in the set. Machine learning typically involves two phases: (1) training and (2) prediction. As explained below, both phases require that the text in each document is processed. Figure 1 illustrates the progression of each phase.

Figure 1. Two-Phase Machine Learning Process



In the training phase of the machine learning process, a supervised learning algorithm generates a predictive model based on a training set. To create a training set, an individual simply compiles examples of relevant and not relevant documents.

In the prediction phase, the predictive model generates predictive scores for all the documents that were not reviewed in the training phase. The predictive score ranges from 0 to 1, with 0 being most likely irrelevant and 1 being most likely relevant. There are many scores between 0 and 1, as each document represents its own potential for relevance. The predictive model's effectiveness is typically measured with the help of a validation set, a representative sample of the overall data set that has been reviewed by an attorney to confirm the relevance of documents in the sample. The predictions generated by the predictive model for the documents in the validation set are compared to the attorney's relevance decision.

Two main statistical measures describe the success of the model: recall and precision.

- **Recall** is the percentage of all relevant documents identified by the model.
  - Recall measures the percentage of relevant documents that the model discovers in the data set and helps answer the question: Did we identify and produce all the relevant documents?
- **Precision** is the percentage of documents the predictive model indicated were relevant that actually were relevant to the document review task at hand.
  - Precision helps to determine which documents to review first and how many irrelevant documents will require review to identify the relevant documents.

Recall and precision are calculated using a cut-off predictive score that separates the likely relevant documents from the likely irrelevant documents. Remember, the predictive model generates predictive scores ranging from 0 to 1, with 0 being most likely irrelevant and 1 being most likely relevant. An example of the relationship between recall and precision illustrates how the two metrics relate to one another. Imagine there are 10 relevant documents in a 100 document data set, and 8 relevant documents have a score greater than .45. Setting the cut-off score at .45 would result in a recall of 8 of the 10 relevant documents in the set, which is equivalent to 80% recall. Also, imagine that in the same total set of documents there are 15 documents that are not relevant but have a predictive score at .45. The precision at this cut-off score is $\frac{8}{8+15}$, which is equivalent to roughly 35%. The precision of 35% means that 3.5 out of every 10 documents identified by the model in this case would be relevant. Recall and precision are usually inversely proportionate measures: as recall increases, precision usually decreases and vice versa. Lowering the cut-off score typically increases the recall but decreases precision. As a result, lowering the cut-off score will require the review of more irrelevant documents.

If the initial parameter settings and training set do not generate an acceptable model, the settings should be modified and training data supplemented to create a new predictive model. This approach is iterative. Once the model achieves acceptable recall and precision rates, it can be used to target relevant content, identify attorney mistakes in relevance decisions, and reduce the volume of documents requiring attorney review.

The Phases of Machine Learning

Preprocessing Phase

Preprocessing is the first step in developing a predictive model, and it transforms the text of the documents into appropriately formatted content for the supervised learning algorithm. The preprocessing phase breaks down the document into smaller units to help the algorithm identify relevant documents. We tested a variety of technical settings in the preprocessing phase to determine which combination would yield the best results. The variables tested in our experiments included both preprocessing parameters and supervised learning algorithms.

The preprocessing phase breaks apart words into manageable units for the algorithm to process. *Tokenization* breaks up the sequence of sentences in a document into a set of smaller units (usually words) called *tokens*. *Token Filtering* removes irrelevant words such as stop words (e.g., a, the, it), numbers, short words (words with just one or two characters), and long words (words with more than 20 characters). *Stemming* converts words into their root forms. For example, the base “stem,” can be used to search for “stems,” “stemmer,” “stemming,” and “stemmed.”

The model then generates n-grams – a sequence of consecutive words (tokens) in a document – which can differ in meaning when combined, such as “black” vs “black market” or “short” vs “short sighted.” An n-gram can be made up of one word or several words. The *N-gram Generation* step generates all words in a document as “features” of the document, which is a value assigned to a particular word or group of words. Features are the values that the machine learning algorithm uses as inputs to assign a predictive score between 0 (likely not relevant) and 1 (likely relevant) to each document in the data set. The *Feature Selection* step applies an algorithm to the training documents to identify a subset of the most effective words (or other features of the document) to represent the intended purpose of the machine learning exercise, such as finding relevant documents or privileged documents in the data set.

The last preprocessing step, *Vector Generation*, transforms the text of a document into a "vector of feature values," or numerical representation of that document.

Predictive Modeling Phase

With the preprocessing phase complete and a vector of feature values established for each document, the supervised learning algorithm can generate a predictive model using the selected words or n-grams from the training set of documents. In a linear predictive model, a weight is assigned to each selected word or n-gram, establishing its ability to discriminate between relevant and not relevant documents. The significance of a word or n-gram to a document's classification as relevant or not relevant is based on two attributes: (i) the weight of the word or n-gram and (ii) the word or n-gram's value. These experiments manipulated different token (word) values and weights – among other technical settings in the model – in order to test the impact of certain preprocessing parameters on the outcomes of the predictive model.

Preprocessing Parameters and Machine Learning Algorithms

The choice of preprocessing parameters and supervised learning algorithm can have a significant impact on the results of the predictive model. Our study analyzed a variety of implementations of the following preprocessing parameters and supervised learning algorithms to test the impact of backend technical settings on predictive models:

- N-Gram
- Token Value Type
- Number of Tokens
- Down Sampling
- Support Vector Machine, Logistic Regression, and an implementation of Latent Semantic Indexing to categorize documents

We begin by using the bag-of-words approach to represent a document as a vector of feature values (a numerical representation of a document). The text is represented as a "bag" of all of the words it contains, disregarding grammar and word order but tracking repeated words. Each word (or token) is counted. This technique simplifies the representation of the text within the document population.

N-Gram

An *N-Gram* is a contiguous sequence of tokens from the text of a document. When the n-gram parameter is *n*, all 1-grams (one word), 2-grams (two words), 3-grams (three words), and so forth, are generated for each document. The n-grams parameter allows a supervised learning algorithm to assess the impact of any combination of words on a review category such as relevance or privilege. For example, words like “station” and “wagon” have very different meanings individually than “”station wagon.” The n-grams parameter provides an opportunity to take such complexities into account. In the “station wagon” example, using 2-grams would generate three tokens: “station,” “wagon,” and “station wagon.” N-grams are established for all documents in the data set.

Token Value

A *Token Value* is assigned to each of the tokens (words) generated from the document. We tested four different types of Token Values in this experiment: binary, term frequency, normalized term frequency, and term frequency-inverse document frequency.

A **binary** value is the most popular type of token value—the word either exists in the document or it does not. If a word occurs in a document, the binary token value is 1; otherwise, it is 0.

Term frequency measures the number of times the word occurs in a document. Sometimes a word’s frequency can indicate its relevance. For example, an article that mentions “Michael Jordan” one time may or may not be about Michael Jordan. An article that mentions “Michael Jordan” a dozen times, however, is more likely to focus on the basketball star.

The third type of Token Value is **normalized (augmented) term frequency**, which helps to ensure that words that occur less frequently in a document are not overshadowed by frequently occurring words. The theory is that not all frequently used words are effective at defining the category. For example, a press release from a company called “AnyBrand” may mention “AnyBrand” many times, but that does not mean the press release is only about “”AnyBrand.” The press release may focus on quarterly earnings but use the phrase “quarterly earnings” fewer times than “”AnyBrand.” In this example, “quarterly earnings” is more effective at distinguishing the press release’s content than “”AnyBrand.”

To further illustrate normalized frequency’s function, assume that “AnyBrand” is the most frequent word in a document, occurring 10 times, and ”quarterly earnings” also occurs twice within the same document. Using the term frequency value type, “”AnyBrand” is considered five times more important than “quarterly earnings” ((10/2) = 5). Using normalized frequency, however, the value of “AnyBrand” is: 0.5 + 0.5*(10/10) = 1 and the value of “quarterly earnings” is: 0.5 + 0.5*(2/10) = 0.6. “In sum, “AnyBrand” is still more important, but not five times more important.

The last token value type is **term frequency-inverse document frequency** (TFIDF), which assigns a value that correlates to the estimated importance of a given word in an individual document. The TFDIF value compares whether a word is frequently used in one document or across the document population. If a word is a very common term such as “place” or “thing” and appears in a large portion of documents in the population, it should have less impact on the model than a word that appears frequently in only one document.

Number of Tokens

Training documents may contain millions of different words (also referred to as “tokens”), including many irrelevant words. Allowing the model to consider all of these words can reduce the effectiveness of the machine learning algorithm. Information gain is a technique used to weed out ineffective words from the process. The information gain of a given word is generally based on the word’s effectiveness at discriminating between the categories of interest: the higher the discrimination power, the higher the information gain. Using words that are most effective at defining the relevant and not relevant classes to train the model will reduce the statistical noise created by ineffective words. Several studies have confirmed information gain’s effectiveness as a selection criterion for predictive modeling tasks.[20]

With the information gain established for every word in the training set, reviewers can target and select the most effective number of words to include in the set. The *Number of Tokens* parameter simply defines how many of the most discriminating words to use from the training set. Reviewers can then transform the available words in the training set into a narrow and highly discriminant set of words for modeling by combining the results of information gain and an optimized number of tokens.

Down Sampling

The distribution of the modeling category (e.g., between relevance and nonrelevance, privilege and non-privilege) is often unbalanced within the document data set of a legal matter. In an unbalanced data set, the majority class (usually not relevant documents) contains a large percentage of all of the documents, while the minority class (usually relevant documents), contains only a small percentage of all documents in the set. Studies[21] have shown that unbalanced class distributions perform poorly with many supervised learning algorithms. *Down Sampling* is frequently used to alleviate the problems caused

[20] *See* Yang, Y. & Pedersen, J.O., *A Comparative Study on Feature Selection in Text Categorization.* (In Proceedings of the 14th International Conference on Machine Learning (ICML)), 412-420 (1997).

[21] Japkowicz N. & Stephen, S., *The class imbalance problem: A systematic study*, 6 INTELLIGENT DATA ANALYSIS, NO. 5, 2002, at 429.

by unbalanced class distribution. Rather than using the entire set of negative training examples from the majority class, a subset of negative examples is selected, such that the resulting training data is less unbalanced and recall may be enhanced. The down sampling parameter defines the percentage of negative (e.g., not relevant) training documents used to create a model.

Supervised Learning Algorithms

We selected two popular machine learning algorithms, *Support Vector Machine (SVM)* and *Logistic Regression (LR)*, for this study. SVM is widely used to develop text categorization models.[22] Additionally, this group of collaborators also included the algorithm implemented in a popular document review platform in the legal domain. The backbone of this platform's machine learning technology is *Latent Semantic Indexing-based* algorithm.

Experiment Data Sets and Design

In this section, we describe the data sets used in our experiments and the experiment setup. Our experiments were designed to thoroughly evaluate the degree to which important preprocessing parameters influence the effectiveness of predictive models.

Data Sets

Our six data sets were from real legal matters by companies in six different industries. The objective was to identify relevant documents using machine learning. Each data set contained Microsoft Office documents, emails, and other text-type documents. Each data set included a set of training documents and a set of validation documents used to calculate the models' recall and precision rates. Attorneys confirmed relevance decisions by manually reviewing documents in both data sets. The documents within each validation set were randomly selected from each data set. Table 1 provides document statistics for each data set. Projects 1, 2, 4, 5, and 6 have disproportionate ratios of relevant and not relevant documents, and thus have unbalanced class distributions, although their training sets are not as unbalanced except for Project 6. Documents in Project 3 are evenly distributed among relevant and not relevant.

[22] *See, e.g.*, Joachims, T., *Text Categorization with Support Vector Machines: Learning with Many Relevant Features* (In Proceedings of the Tenth European Conference on Machine Learning (ECML), Berlin, Ger.), 1997, 137-142.

Table 1. Data Set Statistics

Document Class Distribution	**Project 1**	**Project 2**	**Project 3**	**Project 4**	**Project 5**	**Project 6**
Training – Relevant	1,126	527	5,743	1,285	1,542	159
Training – Not Relevant	2,897	1,114	6,540	2,715	2,458	3,841
Validation – Relevant	206	292	801	486	641	62
Validation – Not Relevant	1,368	1,298	788	1,114	959	1,538

Experimental Setup

We performed 33,600 experiments for this study, using various combinations of the preprocessing parameter values and machine learning algorithms described above.[23] Table 2 details the experimental values for each parameter.

Table 2. Parameters and Values

Parameters	**Parameter Values**
Word Stemming	Yes, No
N-Gram	1, 2, 3, 4
Token Value Type	Binary, Frequency, Normalized Frequency, TFIDF
Number of Words	1,000, 3,000, 5,000, 7,000, 10,000, 15,000, 20,000, 25,000, 30,000, 35,000, 40,000, 45,000, 50,000
Down Sampling	25%, 50%, 75%, 100%
Machine Learning Algorithm	Support Vector Machine (SVM), Logistic Regression (LR), Latent Semantic Indexing(LSI)

Stop words were removed in all experiments. Both the SVM and LR supervised learning algorithms and their default parameter settings were selected from LibLinear, an open source library for large-scale linear classification. The linear kernel was used for SVM. The parameter settings for the machine learning algorithm implemented in the popular document review platform are unknown.

The training documents from each project generated 5,600 predictive models to test all combinations of the parameters. We analyzed the

[23] *See supra* Part II.

performance of each model experiment using the following metrics: recall, precision, and the percentage of documents requiring attorney review. These metrics were determined by comparing the model's classification of the documents with its corresponding validation set, which gave us a sense of how well the model could actually identify the documents we knew were relevant.

We calculated the results of each parameter's impact using the average of all other parameter settings' precisions and the percentage of documents requiring review at a specific recall rate. Using precision as an example, Project 3 generated 2,800 models using SVM and all other combinations of parameter settings (the SVM Model Experiments) and an additional 2,800 using LR and the same combination of parameter settings used for SVM (the LR Model Experiments). To compare the overall effectiveness of SVM versus LR, we calculated the average precision at specific recall rates for each model (30%, 40%, 50%, 60%, 70%, 80%, and 90%). For both SVM and LR we determined the average precision for each set of experiments. The average precision for each set of experiments allowed us to compare the overall performance between SVM and LR. See an example calculation in Table 3.

Table 3. SVM vs. LR Example Precision Comparison

Parameter Type	**SVM Model 1**	**SVM Model 2**	**LR Model 1**	**LR Model 2**
Stemming	Yes	No	No	No
Number of Tokens	50,000	1,000	9,000	1,000
N-Gram	1	4	1	4
Down Sampling	100%	100%	100%	100%
Up Sampling	200%	0%	200%	200%
Token Value Type	Normalized Frequency	Binary	Normalized Frequency	Frequency
Precision @ 90% Recall	72.24%	56.28%	78.97%	60.79%
Precision @ 80% Recall	80.93%	58.61%	86.62%	69.90%
Precision @ 30% Recall	98.77%	95.24%	97.56%	95.24%
Average Precision	83.98%	70.04%	87.72%	75.31%

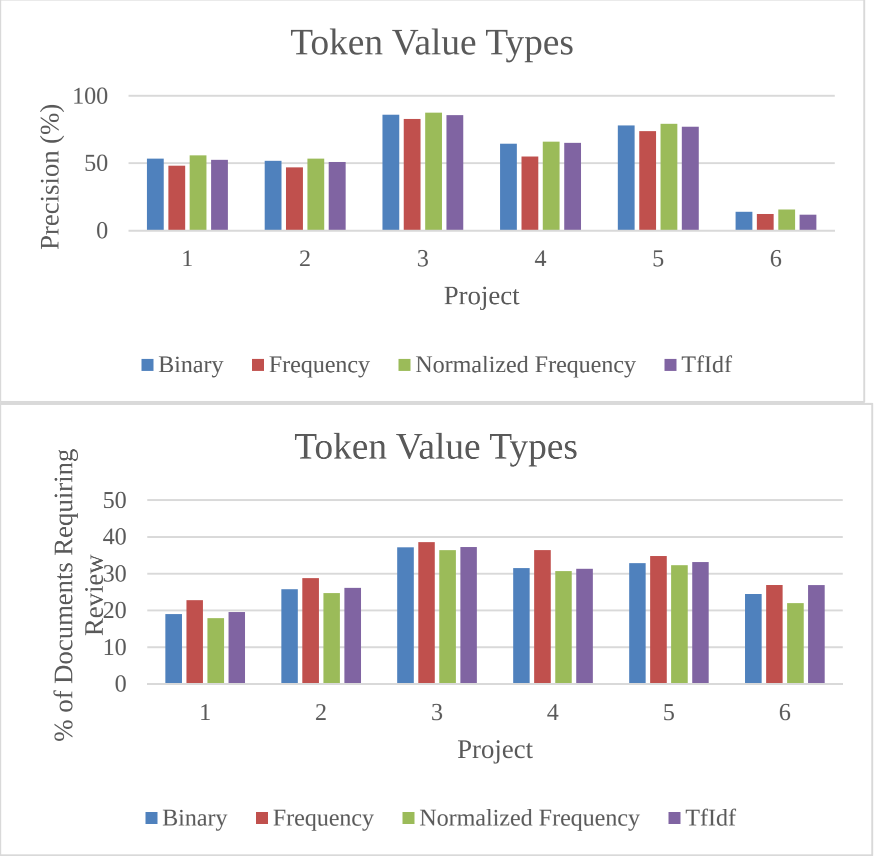
Experimental Results

In this section, we report and discuss our experimental results. For each parameter, we report the average precision and average percentage of documents requiring review at each recall rate. In other words, we report which parameter is most effective at reducing the number of not relevant documents that the model includes in the review set. We calculated the averages using the results of 5,600 predictive models generated for each project using the various combinations of parameter settings. Project 3 had a significantly higher precision rate because its classes were evenly distributed.

Token Value Types

Figure 2 shows the average precision and average percentage of documents reviewed using each of the four different token (word) value types: binary, frequency, normalized frequency, and TFIDF.

Figure 2. Token Value Types



The results of the experiments testing token value types challenge the popular wisdom about how to design predictive models. Both binary and TFIDF token values are widely used to create predictive models.[24] However, normalized frequency performed best in all six experiments by maximizing precision and minimizing the percentage of documents requiring review. Using just the basic frequency value yielded lowest precisions. The binary parameter yielded results in the middle of the pack: while the binary parameter generally achieved better precisions than TFIDF, it did not outperform normalized frequency. Predictive models generated using the binary token value would yield 1.1%, 1%, .8%, .8%, .5% and 2.5% more documents to review for Projects One, Two, Three, Four, Five, and Six respectively, when compared to normalized frequency. While this difference may be small for a limited document review, even one percent inefficiency in a matter with two million documents would result in 20,000 extra documents for review. Like the binary token value, TFIDF is also a popular choice for commercially available machine learning tools. TFIDF is not necessarily effective for predictive modeling, however. Tracking whether a term appears in very few documents does not create meaningful distinctions between relevant documents and not relevant documents. For example, a term occurring in two documents, one relevant and one not relevant, will not indicate the difference between the two.

Normalized frequency's comparative advantage over other token values may stem from the model's ability to adjust the value of less-important words that appear frequently in documents. In other words, as with the AnyBrand example referenced above, normalized frequency reduces the amount of statistical noise in a model. Intuitively, this result makes sense—models that give additional weight to words that appear more frequently can be counterproductive if the tokens used most frequently in a document sample are ones that are less likely to correspond to responsiveness or privilege.

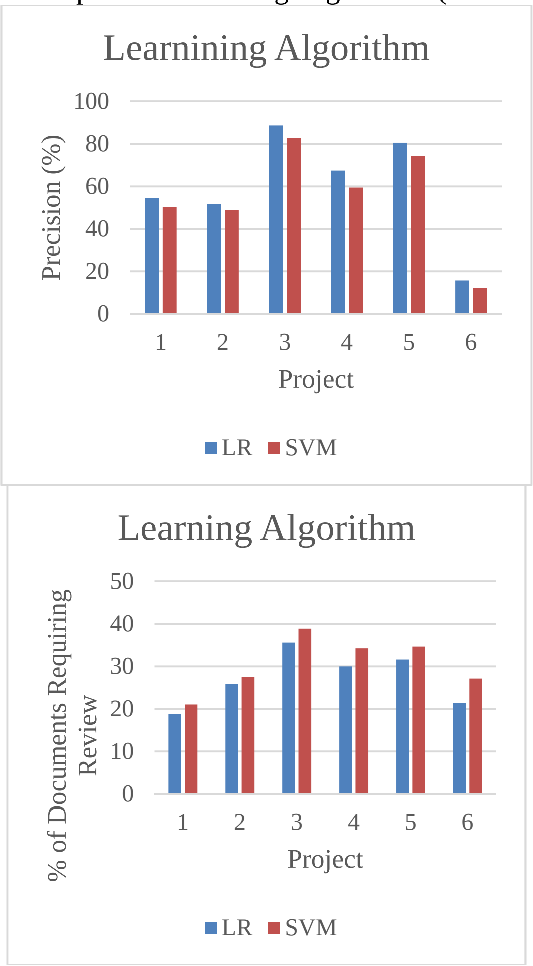
Supervised Learning Algorithms

Our experiments compared two widely-used machine learning algorithms, Support Vector Machine (SVM) and Logistic Regression (LR). Figure 3 displays the average percentage of documents requiring review and average precision for the 5,600 experimental models generated for the

[24] Pascal Soucy & Guy W. Mineau, *Beyond TFIDF Weighting for Text Categorization in the Vector Space Model*, Proceedings of the 19th International Joint Conference on Artificial Intelligence [IJCAI], 1130-1135 (2005).

six projects using both SVM and LR. SVM is regarded as one of the most effective learning algorithms for machine learning,[25] but our results show that LR achieved better results on all three projects. This is true across all recall rates.

Figure 3. Supervised Learning Algorithms (SVM vs. LR)



[25] Yiming Yang & Xin Liu, *A Re-examination of Text Categorization Methods*, Proceedings of theTwenty-Second International ACM SIGIR Conference on Research and Development in InformationRetrieval (SIGIR), 42-49 (1999); *see also* Thorsten Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, In Proceedings of the Tenth European Conference on Machine Learning (ECML), 137-142 (1997).

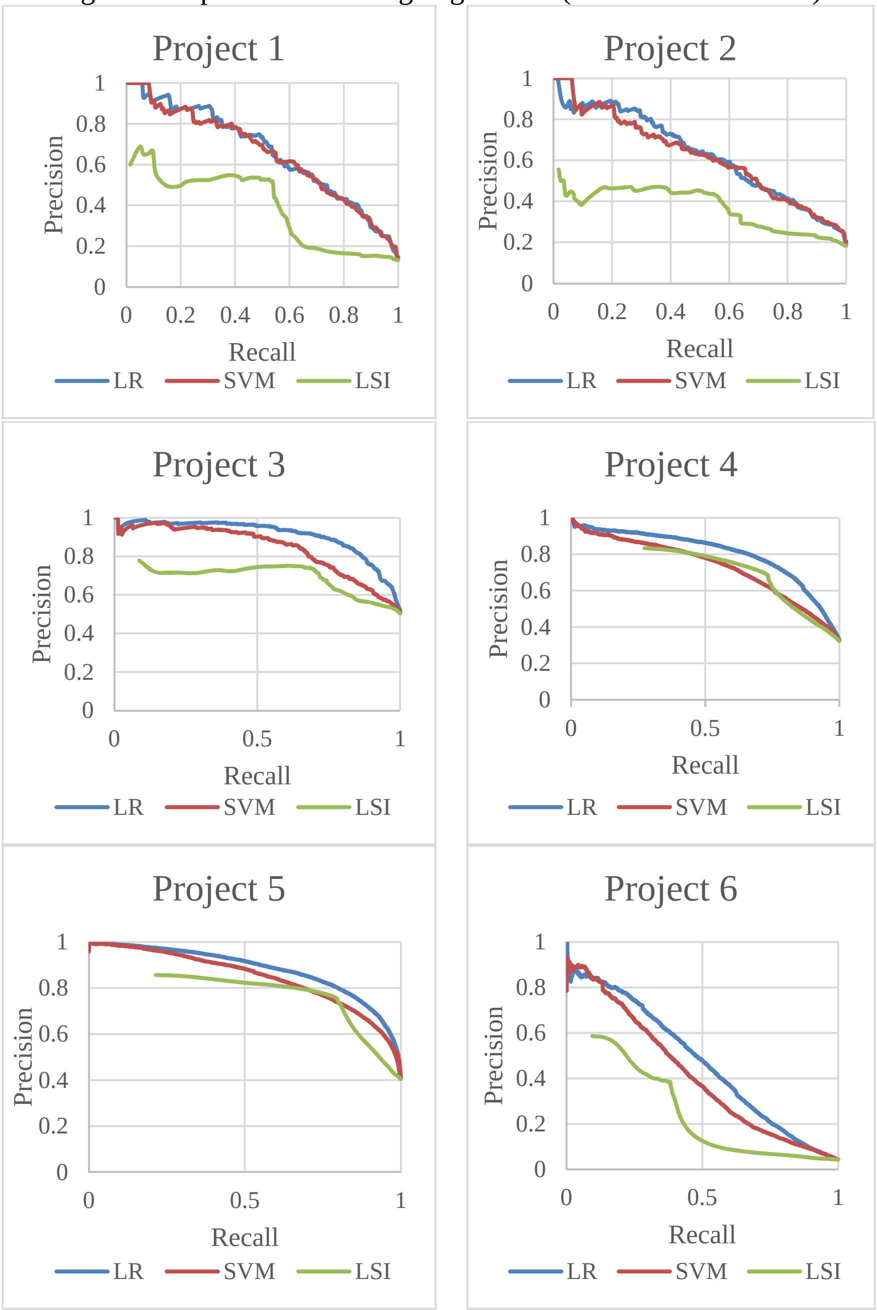
Predictive models generated using SVM would require reviewing 2.3%, 1.6%, 3.3%, 4.2%, 3.1%, and 5.7% more documents than the LR models for each project, respectively. For example, in a matter with two million documents, using a model with the SVM machine learning algorithm would require reviewing an extra 32,000 to 66,000 documents, significantly reducing cost savings.

We also conducted a set of experiments to compare SVM and LR with the LSI-based machine learning algorithm used by a popular legal document review platform. Because parameter tuning is not available in the document review platform, for a fair comparison, we fixed all settings except the Supervised Learning Algorithm when comparing SVM and LR to the LSI-based algorithm. For Project One, Two, and Three, we used the same training and validation documents as listed in Table 1. We also wanted to conduct 'fair' experiments when testing the machine learning application in the popular document review platform. This application typically requires approximately 15,000 documents for training to generate a robust model and because of this, we used more training documents for Projects Four, Five, and Six.

- For Project Four, we used 15,000 documents for training and 136,653 documents for validation, and for;
- Project Five we used 14,983 documents for training and 439,450 documents for validation, and for;
- Project Six we used 15,000 documents for training and 194,550 documents for validation.

The documents used in training and validation are identical across the experiments for all projects when comparing LR, SVM, and the LSI-based algorithm in the popular legal document review platform. In the LR and SVM experiments, we used Normalized Frequency as the token value type, 1-Gram, 20,000 as the Number of Tokens, and no Down Sampling. Figure 4 contains the precision / recall curves of the three algorithms for the six projects. LR consistently outperforms LSI-based algorithm on all projects

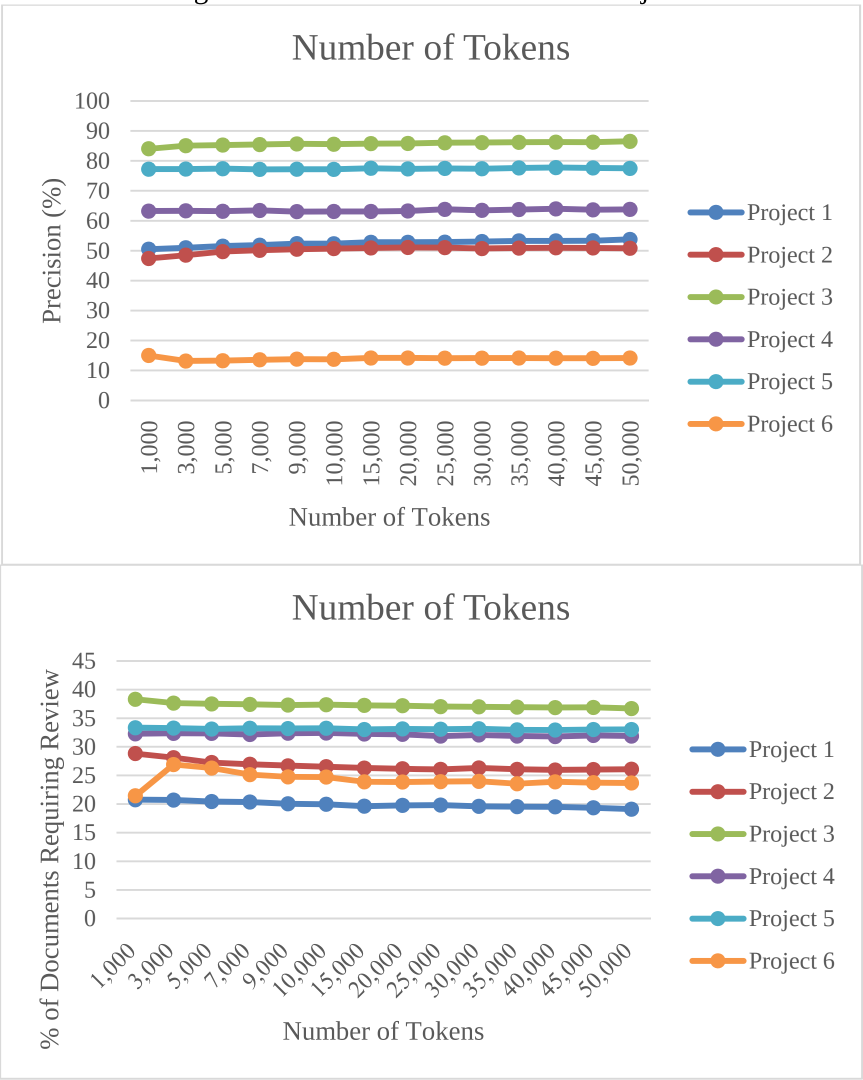
Figure 4. Supervised Learning Algorithms (LR vs. SVM vs. LSI)



Number of Tokens

Figure 5 displays the results of the average precision and average percentage of documents requiring review for the different numbers of words.

Figure 5. Number of Tokens Across Projects

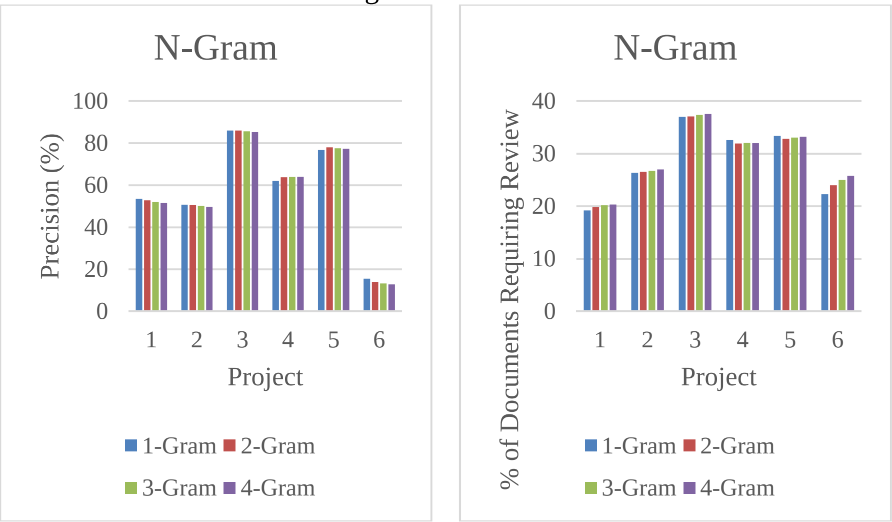


Each model's performance increases as the number of tokens increases. After 10,000 tokens, the rate of increase begins to slow down. We also observed that the effect of changing the number of tokens in each model varies depending on the recall rate. The model's performance improves very little after 10,000 tokens for low recall rates, whereas it increases for higher recalls. This was especially true for Project 3. The model's performance improvement with high recall rates likely occurs because more words are required in order to identify more relevant documents.

N-Gram

As displayed in Figure 6, the average precision decreases generally as *n* increases for the n-gram. 1-Gram (single word) performed the best overall except for Projects 4 and 5.

Figure 6. N-Gram



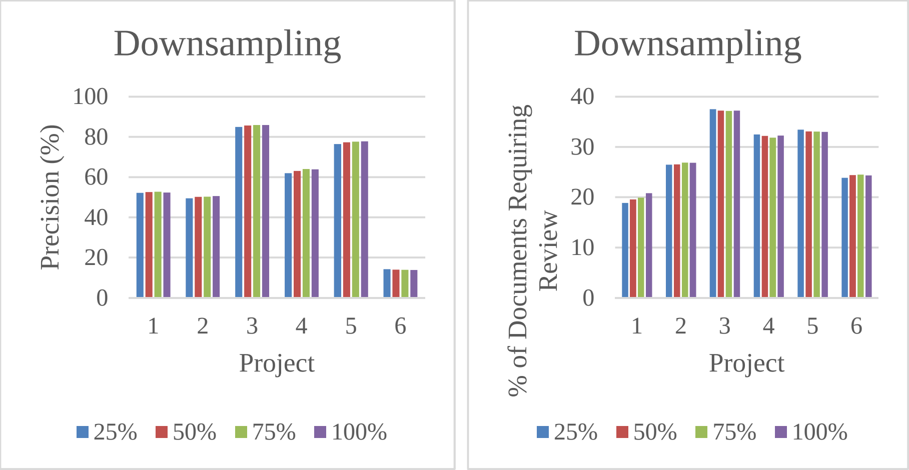
Predictive models generated using 2-Gram, the second best performing n-gram, would require reviewing .6%, .2%, .1%, and 1.7% more documents in Project 1, 2, 3, 6, respectively, when compared to 1-Gram. In a legal matter with one million documents, .5% inefficiency would result in the review of 5,000 extra documents. However, 2-Gram achieved the best performance on Projects 4 and 5 and would require reviewing 0.6% and 0.56% fewer documents, respectively, when compared to 1-Gram.

One possible reason why n-grams become less effective as the value of n increases—i.e. 1-gram is more effective than 2-gram, which is more effective than 3-gram, etc.—is because the model is incapable of distinguishing between meaningful word pairings (like our Station Wagon example) and meaningless ones. In other words, unless all of the specified tokens (words) have a unique meaning when paired with any of the other specified tokens, a model that gives additional weight to word combinations is more likely to generate statistical noise than to increase efficiency.

Down Sampling

There was little variation among the average precision rates for different down sampling values across the six projects, but there were significant differences in the average percentage of documents requiring review for Project 1. Figure 8 displays the average precision and average percentage of documents requiring review for down sampling.

Figure 7. Down Sampling Across Projects



Figures 8 - 13 display the average precision and average percentage of documents requiring review for the six projects at different down sampling percentages for different relevant recall rates. The results show that down sampling has the potential to significantly improve performance at higher recall levels but actually reduces performance at lower recall rates, such as those in Projects 1, 2, 4, 5, and 6. This result is expected for two reasons: (1) the class distribution for these two projects is unbalanced, and (2) fewer not relevant examples allow the learning algorithm to generate a model that is more effective at identifying relevant documents. For Project 3, down sampling negatively affects performance (other than at 90% recall) because its class distribution between relevant documents and not relevant documents is roughly even.

The real-world impact of down sampling on Project 1 is powerful. A model generated using all of the not relevant training documents was 4.2% less precise—and would require review of 4.6% more documents—than a model using 25% down sampling (25% of the original not relevant training documents). In the context of our example of a legal matter requiring review of two million documents, using a model without down sampling would result in 92,000 extra documents for review.

Figure 8. Down Sampling for Project 1

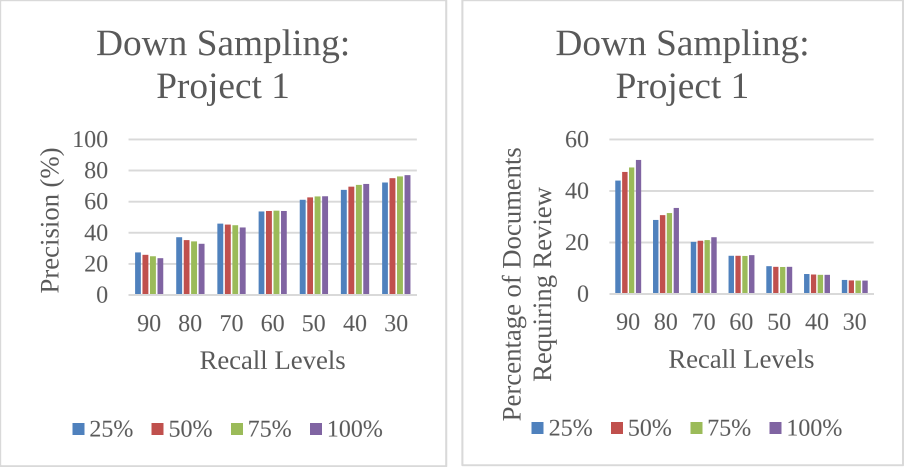


Figure 9. Down Sampling for Project 2

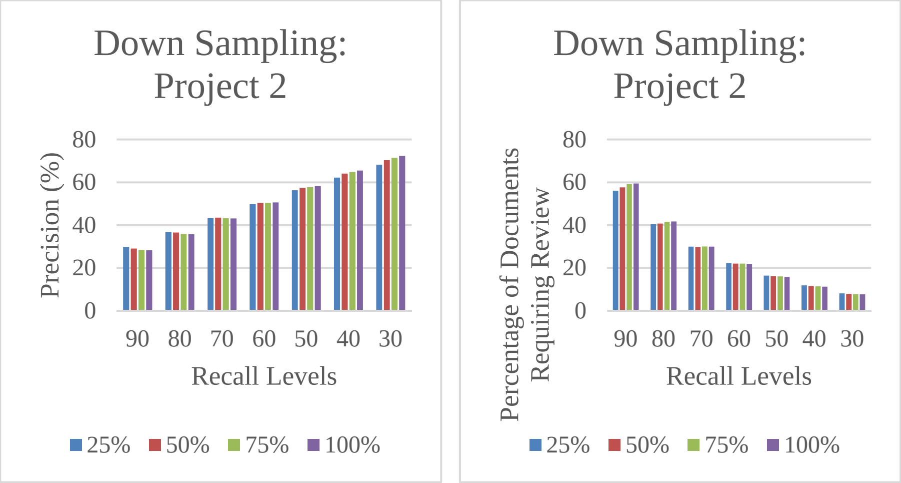


Figure 10. Down Sampling for Project 3

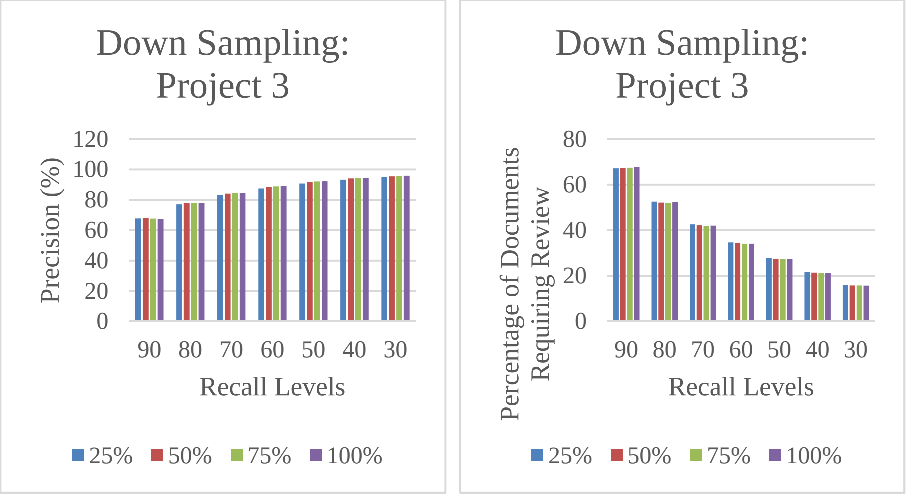


Figure 11. Down Sampling for Project 4

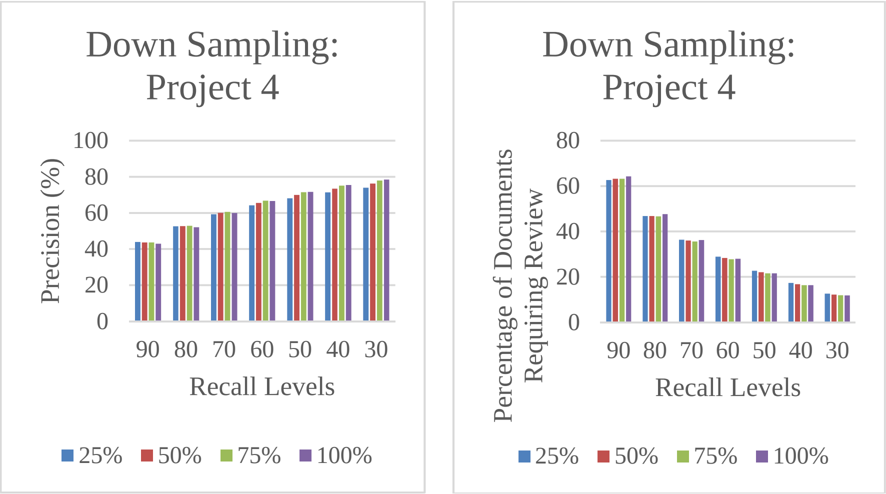


Figure 12. Down Sampling for Project 5

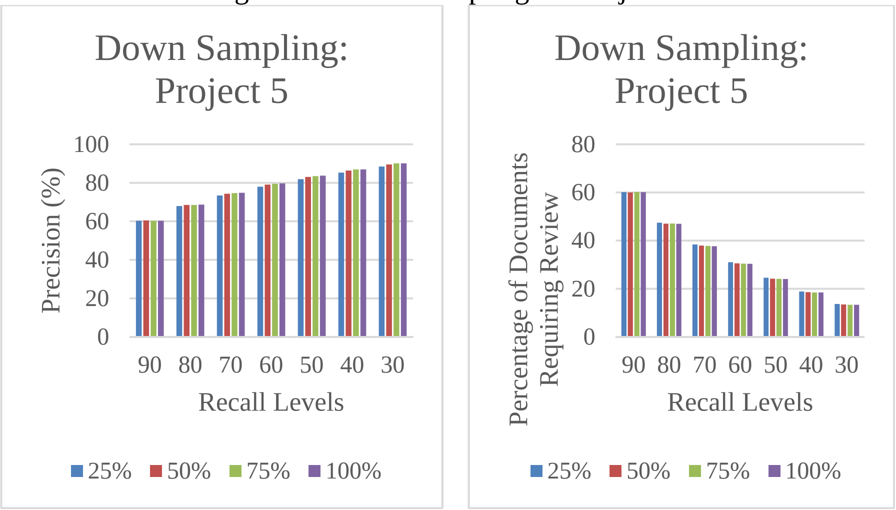
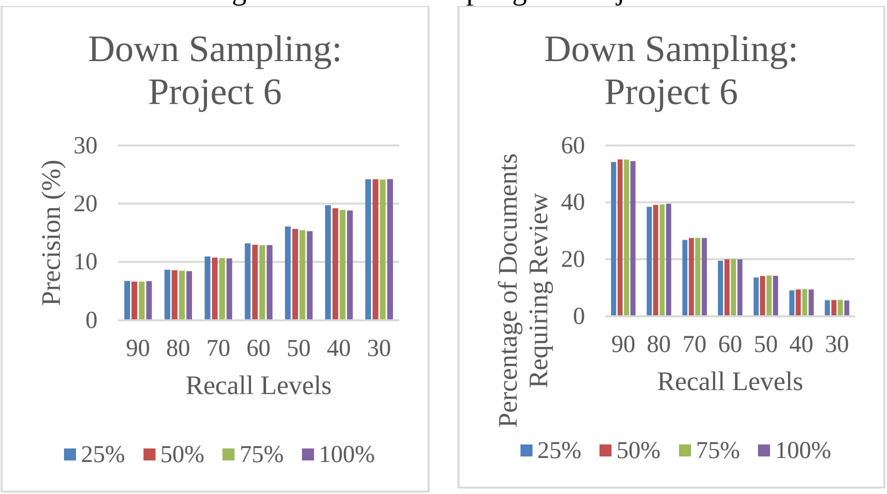


Figure 13. Down Sampling for Project 6



Conclusions

The technology behind machine learning is a black box to most legal teams. Attorneys may, at best, know supervised learning algorithms, but the preprocessing parameter settings used to generate a predictive model are largely a mystery to most practitioners. These practitioners have heard limited information about the impact of preprocessing parameters on the efficiency of the model in large part due to machine learning vendors considering their tools' backend technology to be proprietary and intellectual property. Our study demonstrates that minor adjustments to the preprocessing parameters and the supervised learning algorithm choice can significantly alter the performance of a predictive model.

The results of these experiments showed:

- When it comes to valuing the words (tokens) that appear in a document, the normalized frequency value achieved better results than binary, TFIDF, and word frequency values. Binary, the second best performing value, adds to the review set .9% more documents in comparison to normalized frequency.
- In deciding whether to look at words individually or in groups, the individual 1-gram performed best: the 1-Gram performed better than 2, 3, or 4-Grams. On Project 1, the second best performing n-gram, the 2-Gram, would require review of .6% more documents when compared to 1-Gram.
- In terms of training the algorithm, down sampling performed well with high recall rates on data sets with unbalanced class distributions. In Project 1, the model was 4.2% more precise at 80% recall when a 25% down sampling was applied, compared to a model that used all available, not relevant training documents. However, down sampling did not perform well at low recall rates and on an evenly distributed data set. The results suggest that down sampling should not be used to target relevant documents; only to attempt to drive up the recall on an imbalanced data set.
- Selecting the machine learning algorithm matters. The Logistic Regression algorithm performed much better than the Support Vector Machine algorithm. On Project 3, for example, LR could exclude 3.3% more documents from review. This study also suggests that a popular document review platform's machine learning application, using an LSI-based algorithm as its core technology, does not perform as well as SVM or LR.
- Further, model performance improves as the number of words (tokens) increases, but improvement begins to taper off after 10,000 words are used.

Poor combinations of preprocessing parameters and the machine learning algorithm choice have a sizeable impact on the results of the model. Table 4 shows the results of the best and worst performing models for Project 1. The strongest combination of parameters and algorithms would be 32.18% more precise and would reduce the volume of review by 59.40% in comparison to the worst combination. Note: the LSI-based algorithm was excluded from this analysis because we could not confirm all the preprocessing and backend settings to conduct an unbiased test.

Table 4. Strongest and Weakest Combination of Preprocessing Parameters

Parameter Type	**Strongest**	**Weakest**
Stemming	Yes	No
Number of Tokens	50,000	7,000
N Gram	1	4
Down Sampling	100%	100%
Token Value Type	Normalized Frequency	TFIDF
Supervised Learning Algorithm	LR	SVM
Precision @ 80% Recall	44.84%	12.66%
Percentage of Documents Requiring Review @ 80% Recall	23.38%	82.78%

Our experiments suggest that, on average, the best performing combination of preprocessing parameters and machine learning algorithm to generate a predictive model for a legal matter are:

- Logistic Regression,
- At least 10,000 words,
- 1-Gram,
- Normalized term frequency,
- Stemming and down sampling turned off.
-

The findings explained in this paper look behind the curtain of machine learning for legal teams. While other studies have focused on the results of machine learning as opposed to manual review, this paper has focused on a different question: whether the preprocessing parameters affect the results of the predictive model. As attorneys deploy machine learning for future document review tasks, it is critical to consider preprocessing parameters. Armed with the insights from these experiments, counsel can maximize the precision and accuracy of machine learning models. Both counsel and their clients will save precious resources as a result.