Analysis of US Patent Litigation Outcomes from 1963-2015: A Machine Learning Approach

Hunter Sporn
Princeton University
hsporn@princeton.edu

Joseph Puryear III
Princeton University
jpuryear@princeton.edu

Nicholas Schmeller
Princeton University
nbs@princeton.edu

Abstract

More than 500,000 patents are filed in the United States on an annual basis and most patent lawsuits costs the parties millions of dollars each. The ability to improve the current modeling of patent litigation would greatly improve the performance of the process. With the analysis of US Patent and Trademark Office data from 1963-2015 which contains approximately 74,000 cases of patent litigation, a stronger system can be developed to assist with patent litigation. This paper aims to uncover and explore latent relationships within the existing Kaggle data to determine if a particular case will prove to be a significant burden to the legal system.

1 Introduction

Patent litigation is a "legal process that unfolds when someone who owns the patent for a particular invent enforces their right by suing another for manufacturing or selling the invention without permission" [2]. These processes can become long drawn out processes as both parties compete to justify their action. They are often classified as being long, difficult, and very expensive [2]. The United States legal system has become infamous with regards to its patent litigation. In 2007, the United States "more lawsuits than any other country" with numbers rising every year. [5]. Within the U.S. courts the remedies provided vary from a reasonable royalty compensation to paying above and beyond including the attorney's fees of the other side [5]. However, recent changes to the US patent law system have made patent litigation riskier causing a decline in the number of law suits submitted [6]. The table below provides a brief overview of what patent litigation looks like in the United States in comparison to other companies like Germany and China [6]:

	United States	**Germany**	**China**
Litigation Costs	\$3-4 million	\$100k-500k	\$100k-500k
Time to Trial	24-36 months	9 months	8-14 months
Damages	\$8.9 million median; largest award: \$2.5 billion	Largest award: \$2.7 million	\$17k median; largest award: \$50 million
Discovery	Extensive document and desposition discovery	Almost no discovery	Almost no discovery; private investigators often used to gather evidence.
Contingency Fees	Allowed	Not allowed	Not allowed

Table 1: Patent litigation in the United States, Germany and China [6]

The data set provided comes from Kaggle, a well known online database dedicated to data science. The motivation behind the data set is to explore the "interplay between US patent law and [it's] economic effect", which is crucial to both economic growth and the stakeholders [3]. As described in the "content" portion of the data set listed on Kaggle, "the dataset covers over 74k cases across 52 years" with four different files detailing the "litigating parties, their attorneys, results, locations, and dates" [3]. The largest of the files is dedicated to over 5 million relevant documents.

2 Related Work

Compared to other areas, there exists limited literature with regards to machine learning approaches to patent litigation. However, as machine learning and artificial intelligence continues to expand and develop, their potential applications have been explored in various aspects of law. As mentioned by the Analysis Group, there are plethora of different "new techniques" that "can be harnessed to help attorneys improve legal strategies, conduct informed fact discovery, provide testifying experts with the most complete set of relevant information, and prepare analyses at a previously unseen level of granularity" [9].

One of the most explored aspects of law and machine learning is modeling and predicting the cases that occur at the Supreme Court level. One of the earliest studies of this occurred by Ruger et al. in 2004 which explored the ability to predict the 2002 Term Supreme Court cases based their statistical model compared to typical practices used by legal experts [7]. While analyzing these cases, the statistical model was able to determine the whether the Supreme Court affirmed or reversed a decision 75% of the time compared to the legal experts who collectively got 59.1% [7]. Naturally, the model versus experts varied based on the case's subject and the political views of judges. The statistical model was far better at determining the votes of the moderate judges while the legal experts were better equipped to determine the more left/right leaning judges [7]. However, the difficulty of modeling the lower courts becomes increasing more challenging as additional variations of the judges participating causes more variations in the results. With the development of better machine learning practices, Katz et al. explored the ideas of Ruger et al. further in 2017 by constructing a "model designed to predict the behavior of the Supreme Court of the United States in a generalized, out-of-sample context" from decisions made in cases from 1816 to 2015 [8]. Katz et al. model expands beyond the one term explored by Ruger et al. in their statistical model to develop a larger and more widely applicable model capable of sustaining more variations that results from changes over time like what justices are currently appointed. Katz et al. was able to achieve 70.2% and 71.9% accuracy when predicting the case's outcome and justice's individual vote, respectively [8]. While the models accuracy decreases slightly in comparison to the statistical model used by Ruger et al., the overall consistency and ability to determine the cases across a far larger time period is crucial. The challenges associated with determining the performance of models in law is developing adequate null baseline models to compare them to.

While not specifically targeted at patent litigation, the examination of the Supreme Court cases creates a foundation to build upon. The Supreme Court covers various case topics from tax law to freedom of speech to patent law to administrative law to much more [8]. The ability to accurately predict across a wide variety of cases at the Supreme Court level suggests that the same can occur with a primary focus on patent litigation and could assist lawyers and companies in making the right choices. Pinheiro et al. discusses how "in patent infringement cases [...] machine learning can be used to sort through reams of filings using natural language processing capabilities to reveal features common to desired outcomes" and "such predictions can help the parties decide whether to negotiate a settle or engage in costly litigation" [9]. The efficiency and potential of using machine learning to assist in the litigation process is crucial to reducing the time, costs, and effort associated for both the parties and more importantly the legal system.

3 Methods

3.1 Exploratory Analysis

We use unsupervised learning algorithms to perform exploratory analysis of the data. The aggregated data consist of numerous variables, several of which serve as the targets for exploratory analysis. Specifically, we note that each court handles cases differently. There are a total of 94 federal district courts, the large majority of which processed cases in this data [11]. Each district court is a member of the thirteen federal court circuits. We use the court information present in cases.csv to form a circuit variable, the name of the circuit where the case was tried. Furthermore, we use data on the case cause, demand, jury demand, jurisdictional basis, duration, and number of attorneys for the plaintiff, defendant, counter claimant, and counter defendant parties to identify clustering in the case data. The first four variables are categorical, limited by a select number of categories. The other variables are vectors. Therefore, we find the quartile values of these variables, and modify their val-

ues to one of four values corresponding to their quartile location. Finally, we apply the get-dummies function in pandas to all of the variables, resulting in a dataset with a substantially-increased number of variables, all binary. Our goal in performing this analysis is to better understand how these variables relate to each other and whether the cases can be separated into distinct clusters of cases with common attributes. In clustering, we gain an appreciation for the qualities that make the cases different, which informs our investigative process into the burden each case has on the legal system. We will use principal component analysis (PCA) and non-negative matrix factorization (NMF) to explore the data. PCA is a generative method of dimension-reduction that forms new dimensions orthogonal to each other, sorted by the variance in data. In other words, PCA creates components with maximum variance from the data. We decide to use PCA because it is highly effective in reducing the number of dimensions while retaining information from original data. PCA enables us to identify a smaller basis for the data and reduce noise from a dataset with a large number of variables, such as our modified data set of cases. On the other hand, NMF is a discriminative model that seeks to identify latent variables in the data. By implementing both NMF and PCA, we gain different perspectives on the data at hand. Each algorithm is available for use in the Sci-Kit Learn library. Both PCA and NMF use number of components as a parameter. For PCA, the number of components is the number of orthogonal dimensions assigned to the modified data. We plot the explained variance of the PCA model against the number of components. This plot allows us to choose an optimal number of principal components. On the other hand, NMF is a method of matrix factorization for non-negative factors. We believe NMF is a useful tool for deconstructing this data because many of our variables are strictly positive in nature. For example, the duration of a case varies in magnitude but is always positive. Likewise, no case can have a negative number of plaintiff attorneys or counter-defendant attorneys. Both PCA and NMF prove to be poor algorithms for the data. The NMF model produces a high reconstruction error, while the silhouette score is low and the Davies-Bouldin index high for the PCA models (all internal clustering validation metrics). For that reason, we will use text analysis in exploration of other variables in the data set. In the figure below, we see in (a) a graph showing the distribution of documents by case. A few cases have far more documents than the majority. In (b), we model the total explained variance ratio by number of principal components used in PCA. Finally, in (c), we show a model of PCA clustering with two principal components on the test data derived above with four clusters produced through the K-Means algorithm.

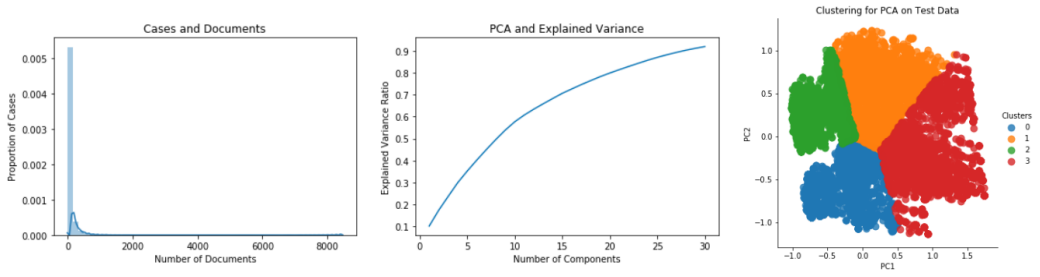


Figure 1: (a) Number of cases distributed by number of documents associated. (b) PCA and Explained Variance (c) PCA Clustering

3.2 Data Set and Feature Selection

The Kaggle patent lawsuits dataset was divided into four separate files: a list of documents for each case, a list of the names of the case and who was represented as plaintiff/defendant/etc., a list of the attorneys associated with each case, and a metadata file containing the timeline of the case, the court the case was heard in, the name of the case, and ID numbers for indexing purposes across datasets.

As mentioned in our introduction section, we are motivated to understand whether there are traits in the metadata of a patent lawsuit that allow us to predict whether it will be a significant burden on the legal system. Based on the information provided in the Kaggle datasets, we decided to represent burden using two main indicators: the amount of time spent on the case, measured from file date to closure date, and the number of documents associated with the case. It is reasonable to assume

that a case that lasts for a long period of time, such as a year or more, is a significant burden on the legal system. It is also just as reasonable to assume that a case with a large number of documents indicates legal resources spent on the case. In fact, this indicator may even be more significant from the perspective of a law office, since while a case can be tabled and rack up a long duration without accruing legal hours, a large number of documents requires personnel resources.

We explored latent relationships to these "burden factors" from:

- a bag-of-words representation of the names of the parties in each case
- a label-encoded representation of the courts and judges involved in each case

3.3 Courts and Judges Relationships with Cases Methods

The primary correlation explored was the association of courts and judges to the duration and document density for each case. Document density is defined as the number of documents associated with the particular trial divided by the duration of the trial in days. This resulted in a density variable of documents per day to compare the amount of effort placed upon the legal system. In order to perform this, the data set was adjusted to accomplish this task by dropping any rows missing either the date filed, date closed, an assigned judge, or the court name. Basic imputation techniques were explored but no feasible operation could be performed that was within the scope of our knowledge without arbitrarily assigning the data. This resulted in a complete data set of 7,288 rows. The length of each case was determined by determining the number of days between the date closed and date filed. The document density was found by taking the total number of documents associated with the case and dividing it by the case duration found earlier. The features used with regards to the duration of the cases was preprocessed using SciKit Learn's LabelEncoder() function [1]. Meanwhile, the features used with regards to the document density was preprocessed with SciKit Learn's OneHotEncoder() function [1]. Upon placing the data in the proper X and y matrix forms, it was split into training and testings states consisting of 80% and 20% of the original data set by using SciKit Learn's train_test_split() function [1]. These data sets were then passed into the following models:

For the case duration, we used the following six models which were implemented using the Python SciKit Learn's Library [1]. The model parameters were left as default unless otherwise stated.

1. Naive Bayes (BNB): Bernoulli implementation
2. Logistic Regression (LR): Fit with 'liblinear' and an auto mutliclass
3. Multinomial NB (MNB): Multinomial implementation
4. K Nearest Neighbors (KNN): With 5 Neighbors
5. Random Forest (RF): Using 100 trees
6. Ridge Classifier (RC)

Likewise for the document density, we used the following five models which were implemented using the Python SciKit Learn's Library [1]. The model parameters were left as default unless otherwise stated.

1. Linear Regression (LiR)
2. Lasso Regression (LaR): Cross validation performed with alphas of 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, and 1e2
3. Partial Least Square Regression (PLS): With 2 components
4. Ridge Regression (RR): Cross validation performed with alphas of 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, and 1e2
5. Support Vector Machines with the Radial Basis Function kernel (SVM): With an auto gamma

The performance associated with predicting the duration of the cases was measured by looking at the precision and F_1 scores. This was easily performed using SciKit's existing function [1]. As described in the documentation for this function, these values are computed as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$F_1 = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = 2\frac{TP}{2TP + FP + FN}.$$

Meanwhile, the performance of predicting the document density was measured using the mean squared error function within SciKit Learn [1]. The lower the value of the mean squared error the better the regression model performed.

4 Results and Discussion

4.1 Bag-of-words representation of case names

The first of our two main approaches to trying to discern which aspects of a case's metadata indicated burden factor well was regression modeling from bag-of-words representations of case names to burden factors. Out of the approximately 72,000 cases present in the sum total of the dataset, only about 9,000 had non-NaN entries for file date and closure date. We chose to trim our datasets to only include cases that had start and end dates. This was ultimately a convenient decision for us in terms of 1) only focusing on data we had without introducing error through imputation and 2) reducing our input to a size that our processor-limited (and cache-limited) laptops could handle.

For data preprocessing, we split our data using the train_test_split() function in Sci-kit Learn along an 80/20 ratio with random selection. We assumed that the randomness would remove any sort of noisy structure that might have been present in the order of cases in the original datasets, and thus we did not undertake to cross-validate. We represented case names in a bag-of-words format using the TfidfVectorizer function from Sci-kit Learn. We varied the minimum word frequency threshold in order to understand which kinds of keywords in case names contained the strongest signal.

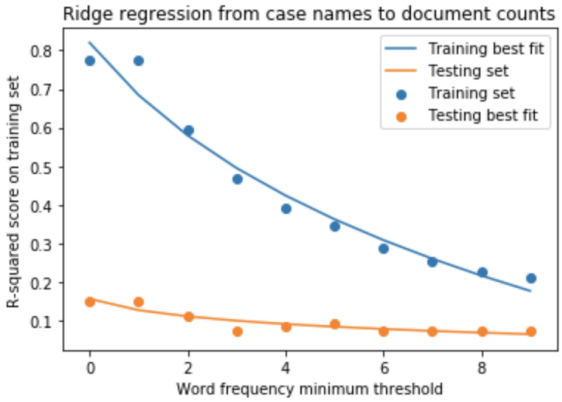
The first regression model we chose to apply to this particular problem was ridge regression (through Sci-kit Learn). While ridge regression is conceptually similar to least-squares linear regression, it is useful for cases where multicollinearity is strongly present, and it has the added bonus of being much less computationally intensive for large datasets (such as bag-of-words for large inputs) compared with ordinary least-squares linear regression. In basic least-squares linear regression testing, we detected indicators of collinearity, specifically a large standard deviation in regression coefficient values as well as varying signs [10]. Given this evidence, we decided to run extensive tests with ridge regression as our "de facto" learning exploration model.

The first relationship we looked to understand was the correlation between specific words in case names with the number of documents involved in each respective case. In this analysis, we treated R-squared values produced from the Sci-kit Learn score() function as accuracies, and we plotted accuracies on the training and testing dataset over minimum word frequency thresholds, shown in Figure 1. We chose to plot best fit curves for R-squared accuracy of the ridge model against word frequency thresholds to attempt to understand the formal connection between model accuracy and word frequency threshold for bag-of-words construction.

The accuracies for the test set in Figure 1 are not desirable for practical application to the legal system. Motivated by these lower scores, we decided to refactor our approach to measuring "burden factor" by changing our metric to document count per day the case was open, in effect a type of document density. We show accuracies of our model across varying word frequency thresholds in Figure 2.

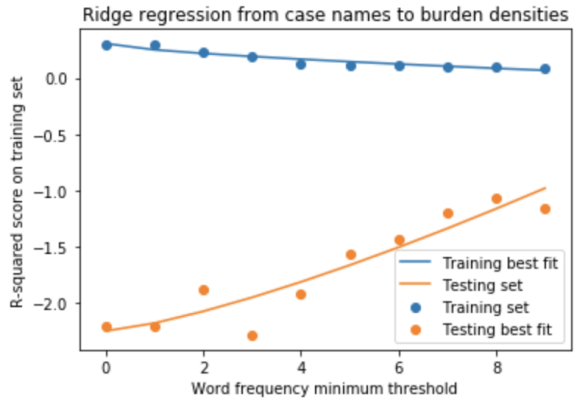
As we can see by the negative R-squared scores for the test set in Figure 2, document density is a much noisier indicator than document counts, to the point where predicting the mean density in the training set is much more likely to be accurate than anything the model can produce. While we initially expected this metric to encode more information that could potentially reveal more latent structures in patent case metadata, the extremely low accuracies made us examine what a density represented more closely. We realized that a single density value can represent many different combinations of document count and case duration: for example, a low density could result from a short case duration with very few documents, or from a very long duration with a high number of documents, or many other linear combinations in between.

With this discovery in mind, we trained one more set of models to predict burden factors from case names. In this case, we decided to solely use case durations as burden factors, and the models we plotted are shown in Figure 3.




```
Function: a * np.power(x, b) + c
Training best fit params:
[-0.4513348   2.84593997  1.29218584]
Testing best fit params
[-0.03710297  0.83363454  0.14899289]
```

Figure 2: Ridge regression model accuracy of predicting document counts per case on training and testing datasets plotted over varying minimum word frequency thresholds, with fitted power law curves. Fits were calculated with the curve_fit() function from the SciPy.optimize library. Ridge regression model imported from Sci-kit Learn linear_model library. np.power() from the NumPy package. Plot generated with pyplot from matplotlib.

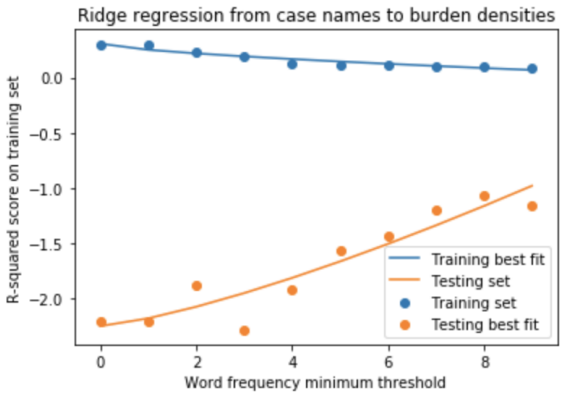



```
Function: a * np.power(x, b) + c
Training best fit params:
[-0.05543041  0.66401291  0.30654797
Testing best fit params
[ 0.07034878  1.31702506 -2.2518408
```

Figure 3: Ridge regression model accuracy of predicting burden densities on training and testing datasets plotted over varying minimum word frequency thresholds, with fitted power law curves. Fits were calculated with the curve_fit() function from the SciPy.optimize library. Ridge regression model imported from Sci-kit Learn linear_model library. np.power() from the NumPy package. Plot generated with pyplot from matplotlib.

While model accuracies for the testing set did not suffer as they did for the densities set of models, the R-squared scores were close to 0 for most word frequencies, indicating that there is not enough signal present in case names to practically predict patent litigation burdens.

Out of these three combinations of document counts and case durations, we show that case names are the strongest indicator for document counts per case. While we were unable to build a model that gave high accuracies for count predictions, we found that the massive disparity in R-squared scores between densities and counts meant that there is cogent, tangible signal present in a case name. Our next objective is to determine whether we can isolate this signal in a way that enables practical applications of our work. We see two ways to determine signal. The first is to build a stronger bag-of-words representation of case names to decrease noise caused by unnecessary words in names being included in our model. We already investigated this by plotting over minimum word frequency thresholds in Figures 1, 2, and 3, but we believe a promising avenue for future work is to rigorously explore parameters that map case names to feature vectors. The second approach we present to predicting case duration with practical accuracy is to group durations and/or document counts into different sets of lengths, e.g. durations that are 0-30 days, 30-60 days, etc. We recognize




```
Function: a * np.power(x, b) + c
Training best fit params:
[-0.05543041  0.66401291  0.30654797]
Testing best fit params
[ 0.07034878  1.31702506 -2.2518408 ]
```

Figure 4: Ridge regression model accuracy of predicting case durations in days on training and testing datasets plotted over varying minimum word frequency thresholds, with fitted power law curves. Fits were calculated with the curve_fit() function from the SciPy.optimize library. Ridge regression model imported from Sci-kit Learn linear_model library. np.power() from the NumPy package. Plot generated with pyplot from matplotlib.

that it is unreasonable for our model to predict exact counts or durations, and we also recognize that practical applicators are not interested so much in exact numbers as they are in determining whether a case will be a lot of work or not. We explore this idea in the next subsection.

4.2 Courts and Judges Relationships with Cases Results

The results of the latent relationship between judges and courts to the duration of their cases are shown in Figure 5. Since numerous cases exist for each judge and court, the challenge of predicting the exact numbers of days resulted in extremely small precision and F_1 scores. In order to overcome this, the cases were placed into bins as described in the figure. These bins allowed for better correlation to the judges and courts and even shown a near linear associations with the size of the bins to the precision and F_1 score. In general, the Naive Bayes Bernoulli implementation performs the best regardless of the bin size for both precision and F_1 score. On occasion, the Naive Bayes Multinomial implement performs similarly to the Bernoulli implement when the precision bin sizes are six months and 1.5 years.

The results of the latent relationship between the judges and courts to the document density described earlier is shown in Table 2. Initial challenges arose when attempting to predict the densities which resulted in all of the mean squared errors being approximately 0.99. It was discovered that this occurred because the majority of the document densities were between zero and one document per day. However, those outside of this range were extremely large towards the range of 196 documents per day making prediction extremely challenging. Various model parameters and features were adjusted in an attempt to resolve this issue, but none were successful. Ultimately, any document density values over one were removed which resulted in 185 cases being dropped across both the training and testing data sets. This improved the performance of all models to approximately 0.02. With this modification to the data set, the Support Vector Machines with the Radial Basis Function Kernel performed better with all the others being nearly identical.

Model	LiR	LaR	PLS	RR	SVM
Original MSE	0.9878	0.9860	0.9878	0.9878	0.9868
Modified MSE	0.0287	0.0287	0.0287	0.0287	0.0272

Table 2: The mean squared error values associated with their respective models when attempting to determine the document density. The original mean squared errors were the results of performing on the initial data set while the modified errors excluded document density values above one.

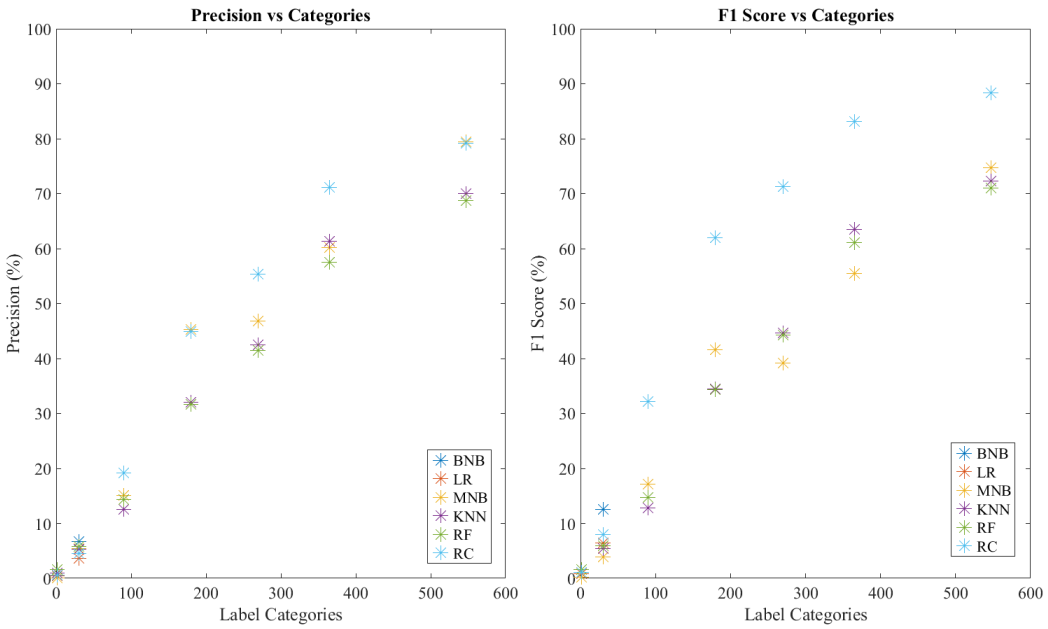


Figure 5: The precision and F_1 score for the various models when exploring the relationship of judges and courts to the amount of time required to process a case file. Label categories segment the days taken for each case into bins correlating to the groupings of a single day, one month, three months, six months, nine months, one year, and 1.5 years.

The results associated with both the duration and document density when focused on the court and judge features provides for initial exploration into how cases burden the legal system based upon time and effort. These results can be expanded upon in a variety of ways. Exploring the relationships by including the Court Circuit the case takes place in would be the first step. This would help segment cases into regions. Another step would be to take further exploration into the cases that were dropped because their document density were too large for the existing models to predict. Perhaps, another relationship exists that would reveal what type of cases are significantly burdensome for the legal system. The ability to further develop these relationships could provide insight into the what cases are burdensome and additionally which courts and judges are either burdened with a lot of cases or are just inefficient.

5 Conclusion

Our two main avenues of analysis in this paper (mapping text analysis of case names and label-encoded court data to litigation burden factors) both produced data that indicates tangible signal in patent data. Because this signal has potentially massive consequences for patent law practice, we offer a few recommendations for future work to improve signal interpretation to a practical strength. One reason we think that accuracies trend downward over word thresholds is that keywords in the name data such as ”engineering” or ”software” or ”insurance” are strongly associated with certain burden factors. While the recommendation of ”analyzing more data” is trite and rarely an effective solution, we believe that high accuracies at low thresholds indicates that more cases with similar names would strongly increase burden prediction accuracy. Throughout our paper’s development, we were also unconvinced that we were using the ”right” models for the data we had, and we encourage further exploration and reproduction of our results with more unique and robust models. From a bird’s eye view, however, we are pleasantly surprised that we discerned such a clear signal in patent metadata, and we hope that more rigorous study continues on this subject.

Acknowledgments

Material throughout this project was inspired by previous implementations and examples provided by COS 424 and our old assignments. One of us discussed ideas about this assignment with Jacob Zimmer and Rachana Balasubramanian (former COS 424 students).

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2011.

[2] What is patent litigation? *Morningside Translations*, 2018.

[3] US Patent Trademark Office. Detailed patent litigation data on 74k cases, 1963-2015. 2018.

[4] Alan C. Marco, Asrat Tesfayesus, and Andrew A. Toole. Patent litigation data from us district court electronic records (1963-2015). *USPTO Economic Working Paper No. 2017-06*, 2017.

[5] Lei Mei and E. Robert Yoches. Unique aspects of u.s. patent litigation. *Lexis Nexis China Legal Review*, 2007.

[6] William C. Spence, Jason Wejnert, and Brian Beck. Global patent litigation strategy. *IAM Media*, 2018.

[7] Theodore W. Ruger, Pauline Kim, Andrew D. Martin, and Kevin Quinn. The supreme court forecasting project: Legal and political science approaches to predicting supreme court decisionmaking. *Columbia Law Review*, 104, 2004.

[8] Daniel Martin Katz, Michael J. Bommarito II, and Josh Blackman. A general approach for predicting the behavior of the supreme court of the united states. *PLOS ONE*, 2017.

[9] Lisa B. Pinheiro, Jimmy Royer, Mihran Yenikomshian, Nick Dadson, and Paul E. Greenberg. Using machine learning in litigation. *Analysis Group Economic, Financial, and Strategy Consultant*, 2017.

[10] Ridge regression. *NCSS Statistical Software*, 2019.

[11] Court role and structure. *United States Courts*, 2019.