Norwegian School of Economics
Bergen, Spring 2020

Predicting Patent Litigation

A Comprehensive Comparison of Machine Learning Algorithm Performance in Predicting Patent Litigation

Henrik Størksen Follesø and Maria Kaminski

Supervisor: Steffen Juranek

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Acknowledgements

This thesis is written as a part of our Master of Science in Economics and Business Administration at Norwegian School of Economics (NHH) within the Business Analytics (BAN) program.

First and foremost, we would like to express our sincere gratitude to Steffen Juranek for excellent guidance and frequent feedbacks throughout the period we have worked on this thesis. We would also like to thank Håkon Otneim for sharing his experience and data calculations with us. In addition we would like to thank Sven Are Nydal and the Norwegian School of Economics for granting us access to high capacity computers at the NHH Studio. Lastly, we want to thank Tone Haaland and Joachim Aae who took the time to proofread and comment on our thesis.

Norwegian School of Economics

Bergen, June 2020

Henrik Størksen Follesø

M. Kaminski

Henrik Størksen Follesø

Maria Kaminski

Abstract

Patents are designed to act as an incentive for innovation by awarding exclusive property rights to the inventor. And as such, patents are one of the main driving forces behind innovation, and ultimately economic growth (Lanjouw and Schankerman, 2004). Patent litigation, the legal process associated with legal disputes regarding patent rights, is hard to predict, surrounded by uncertainty, can be ruinously expensive, and very difficult to insure. Previous research has shown that there is potential for predicting patent litigation, however based on limited data and limited algorithm sophistication.

The purpose of this thesis is to evaluate the extent of which patent litigation can be predicted, what machine learning method is most appropriate, and what are the characteristics that is important for the classifier. The goal is to contribute to reducing the uncertainty that threatens the incentives of innovation by introducing more information through better patent litigation prediction. In particular we focus on the patent litigation insurance market as the most direct application for our research.

This thesis is inspired by the work of Lanjouw and Schankerman (2001) which forms the basis of our research. Building on their work, more data and characteristics are added to the analysis, before other more sophisticated machine learning algorithms are employed and compared. The work relates to anomaly detection, and face similar challenges unique to this area of research.

We find that patent litigation can to a large extent be predicted. Furthermore, adding more characteristics and information increase the predictive power. The largest gains in predictive power stems from the use of appropriate algorithms. Using the *right* algorithm is much more important than using a more advanced or newer algorithm. The Random Forest classifier is found to be the preferred method of predicting patent litigation on our data, as it yields models with high levels of predictive power. We find that patent family size, whether or not the patent is owned by a US company, and the number of backward citations to be the most important characteristics that drives the prediction of litigation.

Keywords – NHH, Master Thesis, Patent Litigation Data, Patent Litigation Prediction, Predictive Analysis, Logit, Random Forest, XGBoost, SVM

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	**Explanation**
AUC	Area Under the Curve
CART	Classification and Regression Tree
FN	False negative
FP	False positive
GAM	Generalized additive model
IPC	International Patent Classification
ISA	International Searching Authority
IT	Information Technology
KNN	K-Nearest Neighbours
LDA	Linear discriminant analysis
NPE	Non-practicing entity
NPL	Non-patent literature
OECD	Organisation for Economic Co-operation and Development
PCT	Patent Cooperation Treaty
PV	PatentsView
QDA	Quadratic discriminant analysis
R&D	Research and development
RF	Random Forest
ROC	Receiver Operating Characteristic
ROSE	Random Over-Sampling Examples
SEC	Securities and Exchange Commission
SMOTE	Synthetic minority Over-sampling Techniques
SVM	Support Vector Machine
TN	True negative
TNR	True negative rate
TP	True positive
TPR	True positive rate
USDA	US Department of Agriculture
USPTO	United States Patent and Trademark Office
WIPO	World Intellectual Property Organization
XGBoost	Extreme Gradient Boosting

1 Introduction

1.1 Motivation and Research Question

Patents are designed to act as an incentive for innovation by awarding exclusive property rights to the inventor. And as such, patents are one of the main driving forces behind innovation, and ultimately economic growth (Lanjouw and Schankerman, 2004). Patent infringement, where an entity is infringing on another entity's protected rights, is meant to be discouraged by the threat of legal action backed by patent law. Patent litigation, the legal process entailing patent rights (not only infringement), can be very expensive and is often regarded as uncertain and unpredictable, as there is a widespread lack of information (Bender, 2000). Insecurity bread by lack of information leads companies to acquire portfolios of defensive patents in order to discourage patent lawsuits (Chien, 2010). Not knowing which patents are most likely to be asserted, or the risk of litigation associated with different characteristics, impede decision making in business, R&D, and new venture (Chien, 2011). The cost of litigation can be severally damaging to companies, and in the worst cases forces bankruptcy. The cost can be so high that companies or individuals with granted patent rights simply cannot afford to enforce them, thus stifling the incentives for innovation. Due to uncertainty and high cost, patent litigation have been practically uninsurable, where insurers must resort to pooled prices far higher than the actual risk, making them impractical in most cases (Bloebaum, 2007; Lanjouw and Schankerman, 2004). We seek to reduce some of the uncertainty by studying the prediction of patent litigation and introducing more information. In a world where intellectual property intensive industries drives large parts of the economic growth, understanding and being able to predict patent litigation is more important than ever (Hagiu and Yoffie, 2013).

Previous quantitative studies have shown that there is potential for predicting patent litigation, however based on limited data and limited algorithm sophistication. Motivated by the increased availability of data, and increase in computational power that allows the use of statistical learning methods on large sets of data, we seek to explore possibilities of predicting patent litigation for decision makers in the intellectual property ecosystem. Therefore, we formulate our research questions as:

> *To what extent can patent litigation be predicted, what are the best methods in doing so, and what are the characteristics that drive the prediction of litigation risk?*

In other words: can the addition of more information, both in terms of more observations and more indicators, improve our understanding of what drives patent litigation risk; can more sophisticated algorithms be employed to improve litigation prediction; what is the most appropriate algorithm for predicting patent litigation; and what are the challenges of predicting patent litigation?

The application for litigation predictions can be many. For patent holders, the stakes regarding patent litigation are high due to the high costs of dispute resolution. More information leads to better decision making. For example, the need for a functional patent litigation insurance market is pressing, however due to little information and high uncertainty, the insurance premiums remain too high to be considered useful for most patent owners as pooled prices are common (Lanjouw and Schankerman, 2004). Better predictions leads to more information, less uncertainty and ultimately a more functional insurance market, which could protect smaller players and preserve the incentives to innovate. For patent portfolio management, litigation prediction might help identifying which patents are at risk, thus reducing cost associated with maintaining large portfolios.

1.2 Structure

Section 2 gives an introduction to patents, patent litigation, applications, previous work, and literature. Section 3 presents the data used in our study, its origins, and structure. Section 4 introduces all supervised classifiers used in our study: logistic regression; decision trees; and support vector machines. The measurement of performance and results are also discussed. Section 5 presents and compares the results and performance of the methods presented in the section 4. In section 6 the performance and characteristics of the preferred model are discussed. Further, applications and future work are discussed. Finally, in section 7, the concluding remarks are presented.

2 Background

In this section we provide a background and overview of patents and patent litigation, applications of patent litigation prediction, and previous and relevant research.

2.1 Patents and Patent Litigation

A patent is a grant of a property right to inventor over an invention, issued by a national patent and trademark office (USPTO, 2020a). Patent grants are effective within the countries of which the national patent offices have issued the grant. Patents have a term, a set period for which the patent is valid, which in the United states is 20 years after the time of application.

A patent can be obtained for technical products, processes, or applications of these if the invention is new, innovative, and useful. The invention must be a practical solution to a problem where the solution has technical character, technical effect and is reproducible. It is not possible to apply for a patent of an idea without explaining or showing how the idea can be implemented in practice (USPTO, 2020a).

Patent litigation is the legal process that involves the rights of patents (Lanjouw and Schankerman, 2004). Patent litigation involves either patent infringement, declaratory judgement, violation of civil rights, or breach of contract (Marco and Tesfayesus, 2017). The process of litigation can potentially be long and expensive, with average damages awards being USD 5.9 million per case, which is why many cases are settled between parties before proceeding to court (PwC, 2018). Even if you win a litigation case, by the *American Rule* (US Department of Justice, 2020), the victor needs to cover its own legal representation. Unless a statutory or contractual exception applies the practice of the American Rule remains the norm in cases involving patents (Maier, 2020).

Due to the large costs associated with patent litigation, there are concerns over its impact on Research and Development (R&D) and especially smaller firms' ability to maintain patent protection (Lanjouw and Schankerman, 2004). In areas where the threat of litigation from larger firms is high, the use of preliminary injunction by larger companies might discourage R&D by smaller companies. Even in cases where there is settlement without suits, the threat of litigation will affect the settlement and consequently discourage R&D (Lanjouw and Schankerman, 2004).

The rise in Non-Practicing Entities (NPE) infringement suits, has contributed to an increasing trend of patent litigation filings. NPEs, which in 2001 brought 144 lawsuits over 578 operating companies, and by 2011 had increased to 1,211 lawsuits targeting 5031 companies, are companies that specialize in the enforcement of patent rights they otherwise do not use (Hagiu and Yoffie, 2013). This increase is mirrored by the growth in patent value driven by the revenues and profits associated with intellectual-property-intensive business. Industries within information and communication technology such as software, semiconductors and mobile communications are particularly exposed to NPE activity (Hagiu and Yoffie, 2013).

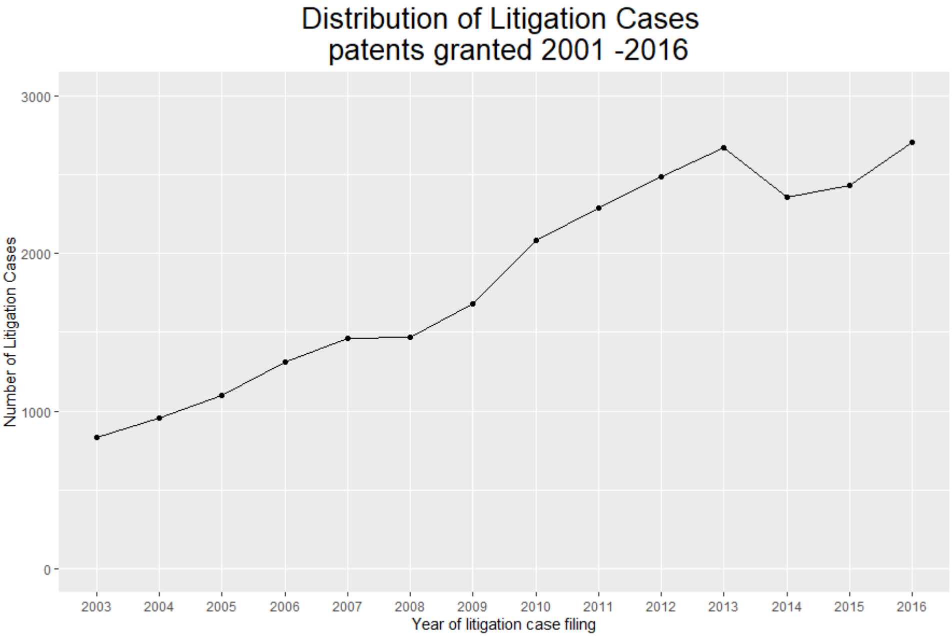


Figure 2.1: Distribution of Litigation cases by litigation filing year, based on data from USPTO (2020c)

From figure 2.1 we see that the total number of litigation cases filed on patents granted between 2001 and 2016 increases each year, except for a slight dip in 2014 and 2015. The total number of litigation cases in the period is probably much higher as these cases are exclusively on patents granted between 2001 and 2016, a limitation introduced by the nature of the availability of our data and is discussed in section 3.1.

2.2 Patent Litigation Insurance

As the consequences of patent litigation for a company or individual can be a very serious, it makes sense to insure against it. However, Lanjouw and Schankerman (2004) describe the patent litigation insurance market as dysfunctional, demand has been severely limited by high prices, while at the same time, the profitability of insurance companies has been undermined by the use of pooled prices and lack of information. In cases where insurance is available, the cost is often prohibitively high. Uncertainty concerning risk factors and inability to effectively predict litigation cases are cited as the main driving force behinds the inefficiency in patent litigation insurance (Lanjouw and Schankerman, 2004).

We argue that increase in information for the insurer will make it easier to price premiums according to actual risk. Lanjouw and Schankerman (2004) found that there is a huge disparity between insurance premiums and actual risk. Premiums can be up to 8 % of the total estimated cost of a litigation process a year, much higher than the estimated risk of litigation that conservatively can be estimated as 1.8 %. The practice of pooled prices reflects the lack of information in the litigation insurance market.

There have been some developments in later years in the litigation insurance market. Coinciding with the rise in patent assertions by "non-practicing" entities (NPEs), there has opened a potential for a different type of patent litigation insurance; *defensive patent litigation insurance* (Ganglmair et al., 2018). As opposed to offensive insurance (or abatement) that covers the cost of enforcing patent rights of the policyholder against an infringer, defensive (or liability) insurance reimburses the policyholder for the cost of defending against allegations of infringement. The popularity of liability insurance has increased in later years, however they remain expensive (Ganglmair et al., 2018).

2.3 Application and Performance Priorities

As our main use-case and application of litigation prediction, we have chosen patent litigation insurance. Therefore, patent litigation is predicted at time of grant in order to be as useful as possible.

The application of the prediction informs they way in which our models should be tuned and trained. In classification, two types of errors can be made. The first is incorrectly classifying negatives as positives, known as false positive error or type I error. The other is incorrectly classifying positives as negatives, known as false negative error or type II error. We would of course prefer not to make any mistakes in predicting litigation cases, but the reality is often that tuning models to decrease one type of error leads to an increase in the other. This is especially true in cases where the frequency of positive cases, such as patent litigation, is very low. Therefore, we need to decide in what direction we want to tune our models, or put differently, what kind of error is worse to make.

One could argue that in the interest of reduced premiums for patent holders, false positive cases should be minimized. That is, if the number of predictive positive cases is reduced, more patent holders get access to the purchase of insurance at lower premiums. For the individual patent holder it is important to get access to insurance at an affordable price.

One the other hand, it is in the interest of the insurer that the number of false negatives is minimized. An insurer would normally prefer to not sell an insurance at all than to sell a cheap insurance to a patent holder that is going into litigation. Thus, the reduction in false negative predictions is important for any given insurer.

The decision of which error type to concentrate on bears importance on the ambitions of this thesis. Theoretically, all you need to do to minimize false positives is to predict everything to be negative. However, this approach does not require any information and does not yield any new insights or information, and will more or less lead to pooled prizes. In our opinion the best approach is to tune our model to the use-case of the insurer as we believe it to be of more value in a business application, in addition to being more interesting to model. We will therefore focus our efforts on increasing the predictive power while minimizing the number of false negative predictions.

2.3.1 Alternative Applications

In addition to the improvement in insurance decision making and pricing, the prediction of patent litigation can provide value for a number of users and applications.

Successful prediction of patent litigation can be of value to law firms representing patent holders and their patent portfolio. Both in terms of deploying resources on patents that are at risk, and in terms of service pricing. Pricing of legal services might be influenced by risk as portfolio management costs can be reduced by concentrating resources on patents that are at risk. This also applies to in-house management of patent portfolios in companies. Firms might have ownership of thousands of patents, and knowing which ones are at risk might improve enforcement efficiency and cost.

For economist, scholars, and policymakers improved predictions on patent litigation a better understanding of risk factors might inform policy changes such that the incentives of innovation are preserved. Moreover, patent litigation is costly for both the parties involved and the society as a whole. Increased understanding of litigation risk of each patent at date of grant cant inform patent officers in their approval of potential patents. An understanding of what drives litigation risk and what patents have increased risk of litigation might lead to the improvement of those patents before they are even granted.

We also hope that some of our research might contribute to the field of anomaly detection, as we seek to find the most appropriate methods for predicting on a rare event such as patent litigation.

2.4 Literature Review

2.4.1 Characteristics of Patent Litigation: A Window on Competition

In the article "Characteristics of Patent Litigation: A Window on Competition" (Lanjouw and Schankerman, 2001) the authors assess characteristics of litigated patents by combining data from US district courts and detailed information from the US Patent and Trademark Office (USPTO). The article provides a broad-based statistical characterization of patent cases filed in the United States, and the authors find that there is strong correlation between a variety of characteristics of the innovation, patent, and owner. Lanjouw and Schankerman (2001) state that these findings are consistent with existing hypothesis in literature. Observable characteristics of patents and owners are systematically related to the probability of law-suits, which according to the authors can facilitate the development of the private litigation insurance market, which could be particularly important for smaller patent owners.

Some of the key findings relate to the nature of the ownership of the patent, the number of citations, claims, and the number of other patents owned by the owner. Lanjouw and

Schankerman (2001) states that a patent is significantly more likely to be cited by other inventors shortly after it is litigated as compared to other patent of the same age litigated longer ago. Another key finding is that individuals are more likely than corporations to go into litigation. Furthermore, some industries and technologies are more likely to involved in litigation than others. For example the most valuable drugs and health patents have an estimated probability of litigation over the lifetime of the patent of more than 25 %.

We expect to find similarities in variable importance, but also seek to expand the business application to the litigation insurance market by providing methods to predict litigation at time of grant.

In their econometric analysis Lanjouw and Schankerman (2001) have employed a binary logistic regression classifier (logit) on a under-sampled dataset. We expect that this type of classifier, although well tested and with low run-times (the time it takes to run a model), is possible to improve upon with newer more advanced methods.

The dataset used in the article is, compared to data available today, fairly limited. We expect that the addition of more observations will lead to increase in classifier performance. Moreover, the number of indicators studied is similarly limited, we expect that the inclusion of new indicators will increase performance. However, where the authors performed regression with full information in hindsight, we are limited to indicators that are known at the time of grant. We expect that there is performance gaps between prediction and regression that will need to be compensated for, due to the loss of valuable indicators. For example, one of the key indicators in the article, forward citations is not available to us as it is not known at the time of prediction.

2.4.2 Other Relevant Work

In the article *Predicting Patent Litigation* Chien (2011) studied various characteristics that influences the likelihood of patent litigation. In contrast to our study, Chien includes traits a patent develops after grant but before litigation. Chien finds that patents that end up in litigation differ markedly from patents that do not. Chien identifies that patents ending up in litigation are more likely to be transferred, re-examined, maintained, and cited, and are more likely to have owners that are different in size and have a loan (Chien, 2011). Presumably, Chien has manually periodised all observations, which is hard to do on a dataset the size of ours, but manageable with the 659 positive cases in Chiens study. Petherbridge (2011), Kesan et al. (2011), and Klabjan et al. (2017) discuss and present weak-points and limitations of Chien's work in both usage perspectives and methodology. Chien has, as Lanjouw & Schankerman, opted to use standard logistic regression (logit) in the econometric analysis.

Klabjan et al. (2017) seek to predict litigation likelihood and time to litigation for patents, which closely align with the objectives of this thesis. The authors test several classification models, such as Random Forest, SVM, and logistic regression. The data used is segmented by technology key-words such as "Wireless Network", "Advertising", and "Telecommunications". Time-to-litigation labels are included corresponding to the number of years between grant and litigation. The best model is found to be a *cluster with ensemble method* which includes financial data sourced from the Securities and Exchange Commission (SEC), and segmented into the "Wireless network" category.

The financial variables utilized include revenue, earnings per share, and market share price. From what time these SEC variables are sourced is not made clear in the paper. The authors state that there is no obvious conclusion with respect to cluster with ensemble method compared with more conventional classification methods. Klabjan et al. (2017) do not elaborate in detail the particular use-case for their predictions, which is also reflected in their ambiguous time of prediction, and inclusion of indicators sourced at different times, including the number of years to litigation in itself. Klabjan et al. (2017) use the SMOTE sampling method. This method introduces synthetic data, which is necessary in their studies to due low numbers of observations. Moreover, imputation of missing values is also employed, meaning that the utilized dataset contains synthetic data both from missing value handling and from sampling efforts. In our thesis, due to the large number of observations available the introduction of artificially created observations and values is not necessary.

In their working paper "Measuring Patent Quality: Indicators of Technological and Economic Value" Squicciarini et al. (2013) seek to contribute to the definition and measurement of patent quality. The writers propose a wide array of different indicators of patent economic value. In the paper they collect, calculate, and analyse data that can be considered as indicators of patent economic value. The authors create a database of indicators they find to be useful in measuring patent quality. Although many of the most useful indicators are based on data not known at the time of prediction, the data base of OECD (2020) serves as the backbone of the expansion on the data basis of Lanjouw and Schankerman (2001).

In the article "Protecting Intellectual Property Rights: Are Small Firms Handicapped?" Lanjouw and Schankerman (2004) study determinants of patent suits and settlement during 1978-99. Regression is performed on a dataset sourced from the USPTO to determine which of the factors studied are important. Their key findings are that litigation risk is closely tied to patent owner's patent portfolio size, and that small companies are at a significant disadvantage in protecting their patents.

Cremers (2004) studies determinants of patent litigation in Germany, and largely supports the findings of Lanjouw and Schankerman (2001). Interestingly, Cremer finds that, in contrast to Lanjouw & Schankerman, that companies have significantly higher probability of patent litigation than individual patentees. Cramer points to differences in the litigation system in Germany and the US may create different incentives, with differences in cost rewarding rules and damage calculation. These findings show that differences and changes in the litigation system can affect the impact and importance of different characteristics. As such, we expect that our findings might differ from older research, as the system, processes or incentives might have changed over time.

Our work is also related to the anomaly detection task reviewed by Chandola et al. (2009), as the number of positive cases of litigation (anomalous items) is severely imbalanced. Imbalanced datasets and anomaly detection present unique challenges for both training and measuring the performance of models. We hope that some of the insights on which methods and practices works best on our data, can contribute to the discussion on anomaly detection.

In contrast to previous work, we combine large number of observations and high number of indicators with machine learning algorithms. This comes at a cost of the ability to create indicators manually (such as stock market information), but we believe that a large number of observations is key to predictive performance in patent litigation. Moreover, in contrast to both Chien (2011) and Klabjan et al. (2017) we seek to predict litigation at time-of-grant by using what Chien (2011) refers to as intrinsic variables. This limits the number of available indicators to employ, but may make the predictions more useful in a business application.

3 Data

In this section, we present the data used in our research. First, the primary sources of data: the OECD patent quality indicator dataset, the PatentsView database and the USPTO Litigation dataset, are presented. Second, the construction of the base dataset inspired by Lanjouw and Schankerman (2001) is discussed. Third, the construction of the complete dataset is presented. Then, summary statistics, target leakage and limitations are discussed.

3.1 Databases

In constructing our research data base, the data used by Lanjouw and Schankerman (2001) serves as a natural starting point. However, much has changed since 2001, not only in terms of available algorithms and increased computing power but also in terms av availability of data. Programs such as the Open Government Agenda has made vast amounts of data freely available in centralized data bases such as PatentsView (USPTO, 2020)

3.1.1 The Patent Litigation Data

The Patent Litigation dataset contains all recorded litigation cases in US courts between 2001-2016. The dataset has metadata for the litigated cases, including parties involved, filing date, and location of the courts (USPTO, 2020c). The purpose of the dataset is to be a first step to a comprehensive source of data that could be of interest for economists, legal scholars and policy makers on patent litigation (Marco and Tesfayesus, 2017).

The patent litigation dataset (USPTO, 2020c) contains 110,647 litigation cases on 45,768 unique patents granted between 1963 and 2016, but the litigation dataset contains only cases filed between 2001 and 2016. In order to make sure that the selection of observations reflects the distribution of litigation cases as closely as possible without introducing bias, we limit patents included in the study to those that have grant date between 2001-2016. The median time between grant date is roughly 3 years. To exemplify why it might be problematic to include patents granted outside of the litigation range, consider a patent granted in 1996 which is litigated in 2001. Most patents granted in 1996 that are going to be litigated are already litigated by 2001, meaning that the patent is just slower to be litigated than the rest and not representative of all patents granted in 1996. Therefore, the range is limited to patents granted 2001-2016. Of the 45,768 unique patents in the litigation set, 25,840 is within our period of study and kept after patents granted outside our time frame are dropped. Patents litigated multiple times are counted once, as to not distort the characteristics of patents which at some point are going to be litigated.

3.1.2 OECD Patent Quality Data

The data from OECD Patent Quality indicator dataset (OECD, 2020) consists of patents and patent applications filed between 1976 and 2019, and are made up of a series of quality indicators developed by Squicciarini et al. (2013) ordered in 15 descriptive columns. The indicators are aimed to capture the technological and economic value of the patented inventions. The proposed measures are based on existing literature and on the information in the documents of each patent (Squicciarini et al., 2013).

3.1.3 PatentsView

PatentsView is a patent data visualization and analysis platform intended to increase the value, utility, and transparency of US patent data (USPTO, 2020). The PatentsView program is a collaboration between the United States Patent and Trademark Office (USPTO), the US Department of Agriculture (USDA) and several universities and research centers. The PatentsView database that links inventors, assignees, location, overall patent activity, and the contents of the patents itself (USPTO, 2020). The PatentsView database serves as the main source of data that is not included in the OECD patent quality database (OECD, 2020).

3.2 The Base Data

Inspired by the research of Lanjouw and Schankerman (2001) we create a dataset that reflects the objectives of their paper "Characteristics of patent litigation: a window on competition". Lanjouw and Schankerman (2001) built a dataset that included 5,452 patent litigation cases during the period 1975-1991 involving 3,887 US patents. They further supplemented their data with a matched set from the total US patent population.

The indicators chosen by Lanjouw and Schankerman (2001) can be sourced from the OECD Patent Quality dataset (OECD, 2020) and the PatentsView database (USPTO, 2020). Differences arise from the exclusion of acquired characteristics in our study, as we seek to predict from the time of grant. Moreover, we make no distinction between type of litigation, whereas Lanjouw and Schankerman (2001) makes a distinction between infringement suits and invalidity suits. The Base data contains four main indicators outlined below.

Citations

In a patent application, the patentee must cite all prior US patents that are related to the patent in the application (Lanjouw and Schankerman, 2001). Patents will include references in their application in order to give context to the nature of the invention, which includes listing possible patents, scientific work and other sources of knowledge at the basis of the invention (Squicciarini et al., 2013). The number of patents cited in the application is referred to as *backward citations* (Lanjouw and Schankerman, 2001). A patent's citations are used to assess an inventions patentability and forms the basis of the legitimacy of the claims stated in the patent itself (Squicciarini et al., 2013). Lanjouw and Schankerman (2001) included forward citations, defined as the number of patents that refers to the patent in question. As this number is not known at the date of grant, it is excluded from our dataset. The number of backward citations is included as a variable in our base dataset and sourced from the OECD patent quality indicator database (OECD, 2020).

Claims

Claims in a patent define the property rights provided by the patent (Lanjouw and Schankerman, 2001). The claims in the patent are usually divided into principal and subordinate claims. The principal claims define the novel features of the invention in their most general and broadest form, while the subordinate claims describe these features in more detail. Often, more claims indicate a more complex patent, however, the patentee has every incentive to claim as much as possible in their application. A patent examiner might require the claims to be reduced prior to the patent being granted (Lanjouw and Schankerman, 2001).

The number of claims is included as a variable in our base dataset and sourced from the OECD patent quality indicator data base (OECD, 2020).

IPC Technology Fields

The International Patent Classification (IPC) is a patent classification system that provides for a hierarchy for the classification of patents according to the different areas of technology they relate to (WIPO, 2020a). The IPC classification scheme consists of eight sections with approximately 70,000 subdivisions. The OECD Patent Quality Indicators database includes a *technology field* indicator which is based on the IPC-technology concordance table developed by WIPO in 2010 (Squicciarini et al., 2013). The classes, shown in table 3.1, are used to group patents by main technology fields and is made up of 6 sectors and 35 fields. Using patent classification that indicates the nature of a patent, makes for a seemingly valuable indicator and can help us study the question; does the nature of what is being patented have any effect on the risk of litigation? In our dataset the IPC-technology field is recorded as a categorical variable with 35 levels.

Lanjouw and Schankerman (2001) use a tech field variable of higher granularity, with the 4-digit IPC code, giving 614 subclasses. However, due to the computational limitations in our statistical computing environment, we are not able to include more than 64 categories in a single variable across all models.

Individual, Corporation, and Origin

Lanjouw and Schankerman (2001) include nationality and type of ownership as a variable in their study. The variable is constructed as follow: Patents are classified as domestic (US), Japanese, or other foreign, based on the inventor's or assignee's address. If there is no assignee the nationality of the address of the inventor is used. If the patent is assigned a company or corporation, the patent is classified as such (Lanjouw and Schankerman, 2001).

In some cases there are multiple inventors or multiple assignees which can have different nationalities. Which nationality is assigned is therefore not straight forward. To mitigate this issue, we utilized a calculated share of ownership based on data from USPTO (2020). A patent is given the nationality of the majority of owners.

IPC Technology Fields

1. Electrical Engineering	**2. Instruments**
1. Electrical machinery, apparatus, energy	9. Optics
2. Audio-visual technology	10. Measurement
3. Telecommunications	11. Analysis of biological materials
4. Digital communication	12. Control
5. Basic communication processes	13. Medical technology
6. Computer technology	
7. IT methods for management	
8. Semiconductors	
3. Chemestry	**4. Mechanical engineering**
15. Biotechnology	25. Handling
16. Pharmaceuticals	26. Machine tools
17. Macromolecular chemistry, polymers	27. Engines, pumps, turbines
18. Food chemistry	28. Textile and paper machines
19. Basic materials chemistry	29. Other special machines
20. Materials, metallurgy	30. Thermal processes and apparatus
21. Surface technology, coating	31. Mechanical elements
22. Micro-structural and nano-technology	32. Transport
23. Chemical engineering	
24. Environmental technology	
5. Other Fields	
33. Furniture, games	
34. Other consumer goods	
35. Civil engineering	

Table 3.1: IPC-Technology Fields (Squicciarini et al., 2013)

3.3 Expansion of the Dataset

In order to study whether increased number of variables can increase predictive power, the dataset is expanded. The additional variables added in our dataset are sourced from the OECD patent quality indicator dataset and the PatentsView data base, and are outlined below.

Patent Scope

In creating an indicator for patent scope Squicciarini et al. (2013) utilize Lerners (1994) definition of patent scope. Lerner (1994) states that the scope of the patent is measured as the number of distinct 4-digit IPC subclasses listed in the patent document. Squicciarini et al. (2013) note that there is empirical evidence of patent scope being associated with economic and technological value of patents, and Lerner (1994) observes that the technological breadth of a patents in a firms portfolio significantly affects the valuation of the firm. One could argue that it is intuitive that valuable patents are more prone to litigation as the incentives of any other operator to pursue litigation action are increased. Moreover, the increase in scope in itself, increases the risk of infringement on other patents. Readers interested in how the patent scope is calculated can refer to Squicciarini et al. (2013).

The patent scope is included as a continuous variable and sourced from the OECD Patent Quality Indicators dataset (OECD, 2020).

Citations to Non-patent Literature (NPL)

Most patents include a list of citations. The number of references to other patents are recorded as backward citations, while the references to sources other than patents are referred to as citations to non-patent literature (NPL). NPL often consist of peer-reviewed scientific papers, conference proceedings, databases, and other relevant literature (Squicciarini et al., 2013). The intention is to add references that reflect the prior art that inventions have built upon. Branstetter (2005) finds that patents citing NPL are significantly higher quality than patents that do not cite scientific literature.

NPL is recorded as a continuous variable sourced from the PatentsView database (USPTO, 2020).

Patent Family Size

The Paris Convention (1883) allows patent applicants up to 12 months from the first filing of a patent application to file applications in other jurisdictions on the same invention and claim priority on the date of the first filing (Squicciarini et al., 2013). The number of jurisdictions a patent is filed in and thus related by a common priority filing is referred to as patent family. The Patent Family Size indicator is a measure of the number of patents within the patent family. The size of the patent family is generally associated with the geographical scope of the patent protection. Patents with large patent families are found to be particularly valuable (Cremers et al., 2003). Chien (2011) finds that the patent family size is an important indicator of litigation risk.

The size of patent family is proxied by the number of patent offices at which the invention has been protected (Squicciarini et al., 2013). Patent Family Size is included as a continuous variable and sourced from the OECD Patent Quality Indicators dataset (OECD, 2020).

Originality index

Patent originality aims to describe the broadness of the patent in terms of the number of different technologies cited and to what extent. Meaning that the originality index measures the technology classes of backward cited patents (Squicciarini et al., 2013). The originality measure, proposed by Trajtenberg et al. (1997), invokes the concept of knowledge applications and the importance of cross domain knowledge for innovation. In the OECD dataset the originality indicator is defined as the percent of citations (backward) made by the patent to a patent technology class (Squicciarini et al., 2013).

The originality index is included as a continuous variable and sourced from the OECD Patent Quality Indicators dataset (OECD, 2020).

Radicalness Index

In the OECD dataset, the Radicalness indicator aims to measure the degree to which the patent is different from the patents on which it is based. The index in proposed by Shane (2001) and is a count of the number of IPC classes that the cited patents belong to.

Shane argues that the more different technology classes different cited patents belong to, the more radical the invention must be.

The radicalness index is included as a continuous variable and sourced from the OECD Patent Quality Indicators dataset (OECD, 2020).

Grant Lag

The grant lag is defined as the time between filing data of the application and the date of the patent grant (Squicciarini et al., 2013). Squicciarini et al. (2013) state that there is evidence of an inverse relationship between the value of a patent and the length of the grant period. There might be several reasons as to why some patents take longer time to grant than others, such as patent complexity, family size, and scope. However, studies (Harhoff and Wagner, 2009; Régibeau and Rockett, 2010) have shown that applicants might accelerate the grant procedure for their most valuable patents, by documenting their application well and following the work of the patent office closely. Harhoff and Wagner (2009) found that more controversial claims lead to longer grant lags and that well-documented patent applications are approved faster.

The grant lag is included as a continuous variable and sourced from the OECD Patent Quality Indicators dataset (OECD, 2020).

Assignees

The inventor(s) of a patent are recorded at the point of the filing of the patent. However, in some cases the inventor is not the owner of the patent. In many cases where the inventor is an employee in a company, the patent itself is owned by the company. The company is then *assigned* the patent, meaning that the ownership of the patent is transferred to the company and receives the same rights as that of the original assignee (USPTO, 2020a). In such cases the patent has a recorded assignee and an inventor, in cases where the inventor is the owner of the patent no assignee is recorded. Moreover, one patent may have several assignees, typically according to their stake in the ownership of the patent (USPTO, 2020a).

Utilizing information about assignees lets us explore some interesting questions about the nature of litigation, such as; does having a company or other entity owning a patent increase or decrease the risk of litigation; and does the number of assignees on a patent affect the risk of litigation?

From the assignee section of PatensView (USPTO, 2020) two predictors are created.

- Number of Assignees - Number of Assignees of given patent (0 if no assignee)
- Binary Assignee - A binary variable, 1 if there is one or more assignees, 0 if no assignee

These predictors are not given as data points in the database, but are created through the link between assignee ID and patent ID.

The number of patents per assignee were also considered, and cited by Chien (2011) to be an important indicator. However, the risk of target leakage outweighs the benefit of inclusion, as we are unsure whether we can successfully periodize these observations.

Patent Cooperation Treaty

The Patent Cooperation Treaty (PCT) is an international treaty with more that 150 contracting states, which makes it possible to seek patent protection for a invention simultaneously (USPTO, 2020b). The goal of the PCT is to decrease cost and labour associated with applying for patent protection in multiple nations and jurisdictions.

An interesting dynamic of the PCT is that the process entails an *International search* (WIPO, 2020b). In the process of the international search, an International Searching Authority (ISA) identifies prior patents, published documents, and technical literature, which may have an influence on the patentability of an invention (WIPO, 2020b). Therefore, the PCT process involves an international screening process across multiple jurisdictions. Filing with the PCT might signal a more valuable patent or at least international ambitions for the product.

We have constructed a binary predictor that is 1 if the patent is registered in the PCT system, and 0 if not. The data is sourced from the PatentsView data base (USPTO, 2020).

Foreign Priority

According to the USPTO, foreign patents may be given priority in its application process in the US. Generally speaking, foreign priority is available when there is a previously filed application for a patent for the same invention in a foreign country that afford similar rights and privileges to US patents (USPTO, 2020d). We have created a binary predictor that takes the value 1 if a patent has been given foreign priority and 0 if not. The Foreign Priority indicator is sourced from the PatentsView database (USPTO, 2020).

Lawyer Data

According to the USPTO, only registered patent attorneys, agents, and individuals granted limited recognition, may represent patents applications before the USPTO. Thus, most patents have information about the lawyer associated with the application of the patent. In some cases however, a lawyer is not recorded in the patent data. We have constructed a binary predictor that takes the value of 1 if there is recorded a lawyer in the patent data and 0 if not. The data is sourced from the PatentsView database (USPTO, 2020).

3.4 Descriptive Statistics

In the full dataset there are 3,147,402 observations, 21 independent variables, and 1 dependent binary variable. There are 9 continuous variables and 12 categorical variables. All independent variables, type, and source are displayed in table 3.2. The horizontal line indicate which variables belong to the original dataset and which are added in the extended dataset.

Variable name	Type of variable	Source
Backwards citations	Continuous	OECD PQ
Claims	Continuous	OECD PQ
IPC Technology Field	Categorical	OECD PQ
US Individual	Categorical	PatentsView
US Corporation	Categorical	PatentsView
Foreign Individual	Categorical	PatentsView
Foreign Corporation	Categorical	PatentsView
Japanese Individual	Categorical	PatentsView
Japanese Corporation	Categorical	PatentsView
Patent Scope	Continuous	OECD PQ
Patent Family Size	Continuous	OECD PQ
Originality Index	Continuous	OECD PQ
Radicalness Index	Continuous	OECD PQ
Grant lag	Continuous	OECD PQ
Number of Assignees	Continuous	PatentsView
Assignee Binary	Categorical	PatentsView
Patent Cooperation Treaty	Categorical	PatentsView
Foreign Priority	Categorical	PatentsView
Lawyer Binary	Categorical	PatentsView
Number of NPL Cites	Continuous	PatentsView

Table 3.2: Variables in the full dataset

Table 3.3 shows the distribution both in absolute and relative terms. A point of interest is that only 6.3 % of all patents are owned by US individuals, while 44.4 % are owned by US corporations, meaning that almost half of all patents are owned by foreign (or Japanese) assignees. While almost half are foreign owned, only 13.3 % have filed through PCT.

Distribution of Continuous Variables
2001 -2016

Variable	Min.	1.st Qu.	Median	Mean	3rd Qu.	Max.
Bwd Cites	1.00	7.00	13.0	22.1	24.0	310.0
Claims	1.00	9.00	16.0	17.2	21.0	887.0
Patent Scope	1.00	1.00	2.00	1.96	2.00	31.00
Family Size	1.00	1.00	3.00	3.93	5.00	57.00
Originality	0.00	0.6859	0.8154	0.7559	0.8886	0.9938
Radicalness	0.00	0.1667	0.3571	0.3904	0.5854	1.00
Grant lag	0.00	722	1041	1155	1462	14060
Num Assignees	0.00	1.00	1.00	0.9539	1.00	14.00
Num NPL	0.00	0.00	0.00	5.316	3.00	199.00

Table 3.3

In table 3.4 the distribution of the continuous independent variables are shown. It is interesting to note that in several of these variables there is a huge difference between median and max value suggesting that there are outliers with higher values than most other patents. An example of this is the number of non-patent literature cites which as a median of 0, third quartile of 3 and a maximum of 199. Another point of interest is the 1,041 median grant delay, showing that for most patents it takes almost 3 years from application to grant.

Tech Fields

The distribution of tech field across all patents within the dataset is shown in table 3.5. The most numerous technology groupings are: "Computer Technology (6)"; "Electrical machinery, apparatus, energy (1)"; "Semiconductors (8)"; and "Audio-visual technology (2)". Where by far the largest is Computer technology with 428,740 patents.

Categorical Variables
2001 - 2016

US Individual		US Corporation		Foreign Individual		Foreign Corporation	
1	0	1	0	1	0	1	0
198,281	2,949,121	1,397,240	1,750,162	82,147	3,065,255	836,102	2,311,300
0.063	0.937	0.444	0.556	0.026	0.974	0.266	0.734

Japanese Individual		Japanese Corporation		Assignee Binary		PCT	
1	0	1	0	1	0	1	0
9,235	3,138,167	626,764	2,520,638	2,895,780	251,622	419,567	2,727,835
0.003	0.997	0.199	0.801	0.920	0.080	0.133	0.867

Foreign Priority		Lawyer Binary	
1	0	1	0
1,238,922	1,908,480	2,896,045	251,357
0.394	0.606	0.920	0.080

Table 3.4

Distribution of Tech Fields in dataset

2001 -2016

1 225,567	**2** 202,184	**3** 157,009	**4** 160,481	**5** 64,544	**6** 428,740	**7** 32,320
8 202,915	**9** 148,130	**10** 148,396	**11** 11,810	**12** 44,618	**13** 158,559	**14** 75,630
15 65383	**16** 79917	**17** 46388	**18** 16860	**19** 40924	**20** 30914	**21** 48632
22 1,377	**23** 53,080	**24** 26,486	**25** 66,042	**26** 61,275	**27** 72,929	**28** 53,825
29 60,697	**30** 20,544	**31** 69,367	**32** 105,669	**33** 65,134	**34** 38,280	**35** 62,776

Table 3.5

Table 3.6 shows the distribution of litigation cases by tech field and shows the number of litigation cases within each technology field and the percentage of all patents within each field being litigated. This allows us to assess which fields have high number of cases being litigated as compared to the total population.

Distribution of litigation by tech field

With percentage of tech field category in litigation
2001 -2016

1	**2**	**3**	**4**	**5**	**6**	**7**
1,471	1,218	1,476	1,568	197	3,128	1,032
0.652 %	0.602 %	0.94 %	0.977 %	0.305 %	0.73 %	3.19 %
8	**9**	**10**	**11**	**12**	**13**	**14**
564	531	933	100	687	1,724	421
0.278 %	0.358 %	0.629 %	0.847 %	1.540 %	1.087 %	0.557 %
15	**16**	**17**	**18**	**19**	**20**	**21**
548	1,790	114	180	214	111	249
0.838 %	2.240 %	0.246 %	1.068 %	0.5223 %	0.360 %	0.512 %
22	**23**	**24**	**25**	**26**	**27**	**28**
2	464	238	678	519	255	222
0.145 %	0.874 %	0.899 %	1.027 %	0.847 %	0.350 %	0.413 %
29	**30**	**31**	**32**	**33**	**34**	**35**
785	185	379	992	1,253	675	937
1.293 %	0.901 %	0.546 %	0.873 %	1.924 %	1.763 %	1.493 %

Table 3.6

We see that technology field 6, computer technology, has the highest number of litigation cases with 3128. However, as the number of patents grouped within this field are also high, the relative percentage is a relatively low at 0.73 %. The technology field with the highest percentage of cases being litigated is "IT methods for management (7)" with 1,032 of 32,320 patents being litigated giving a rate of 3.19 %. Pharmaceuticals also achieves a high rate of 2.240 % with 1,790 of 79,917 patents being litigated, which supports the findings of Lanjouw and Schankerman (2001). The lowest litigation rate is achieved in "Micro-structural and nano-technology (22)" with just 2 litigation cases of a total count of 1,377 patents.

Litigation

Of 3,147,402 observations, 25,840 are litigated, giving a event (positive) ratio of 0.82 %. In terms of anomaly detection, such ratio makes the data severely imbalanced (Kuhn and Johnson, 2013). There is an overall increase in the number of litigation cases in our period of study, which can be seen in figure 2.1. Bar a small decrease between 2013 and 2014, there is steady growth of litigation cases filed. There are patents in the dataset that are (as of 2020) classified as not litigated, but that can be litigated in the future. The median delay between the grant and litigation date being about 3 years supports that cause.

3.5 Data Quality

3.5.1 Target Leakage

Target leakage can happen when some independent variables in the dataset include information that is not available at the time of prediction. Therefore it is important to consider when, chronologically, the data becomes available. There are several indicators in the OECD (2020) patent quality indicators that improve the predictive power of our models but are excluded due to target leakage. Chien (2011) refers to information and characteristics of patents that are known at the time of grant as *intrinsic* characteristics, as opposed to characteristics that are attained during the "lifespan" of the patent. For example, the information on forward citations of the patents cannot be used for prediction of litigation at the point of the grant of the patent as no other patents have, naturally, cited the new patent yet. Although the inclusion of variables based on acquired characteristics improve model performance, it uses information that is not available in the intended use-case.

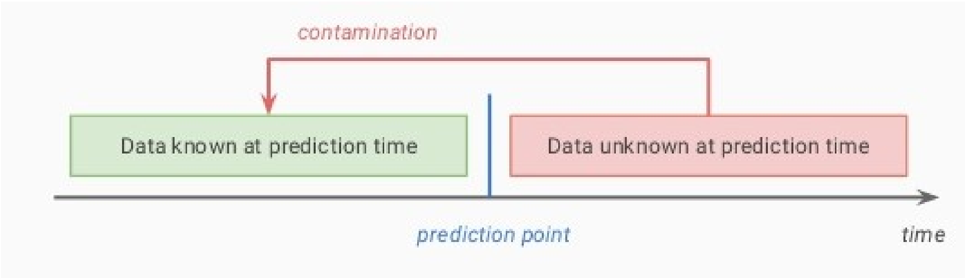

contamination
Data known at prediction time
Data unknown at prediction time
prediction point
time

Figure 3.1: Target leakage (Guts, 2018)

3.5.2 Missing Values

The constructed dataset has some missing values, but on a manageable level. Klabjan et al. (2017) and Chien (2011) cite difficulties in dealing with missing values, which ultimately hamper their results. There are several strategies for mitigating missing values, ranging from imputation [1] to data dropping. As we have a large number of observations available and the missing value rate is relatively low, we opt for data dropping. This implies that if an observation has missing values in at least one of the 21 independent variables, the observation is dropped. Imputation was considered but was deemed more harmful than helpful. Imputation introduced bias through synthetic data, but failed to increase predictive performance.

3.6 Potential Limitations

The dataset and the implications that are drawn from the data are subject to several limitations. According to Lanjouw and Schankerman (2001, 2004) patent litigation data is under-reported, meaning that patent litigation filings and proceedings may not be recorded in the USPTO data base. Thus, the number of litigation cases, and therefore the number of patents that are litigated might be higher in reality. In the worst-case scenario, there might be a specific type of patent or a set of characteristics that are disproportionately represented in non-reported litigation cases. Such that there is an entire group of patents with higher litigation risk that we cannot identify with our models.

Another concern is the disproportionate number of litigation cases in the later grant years, especially in the years more recent than the 3-year median time-to-litigation (Figure 2.1). There are probably many patents yet to be litigated that are currently recorded as not-litigated. Meaning that many patents probably are wrongly labeled. However, the introduction of remedies might introduce bias which we cannot properly control. If we are to limit the dataset by excluding the more recent years, where do we stop? One could argue that a reduction by 3 years (median time-to-litigation) is preferable, however, we have deemed it preferable to include all recent years not only to reduce bias but also to capture more recent development in litigation and patent behavior.

[1] *Imputation* is a technique of estimating missing variable values based on other variable values, typically achieved with averaging values (Kuhn and Johnson, 2013).

4 Methodology

In this chapter we present the methodology of our study. Predicting patent litigation is related to classification and supervised learning. As such, we first give a brief introduction into these topics before exploring the classification algorithms Logistic Regression, Tree Ensembles, and Support Vector Machines. We then present the issue of measuring classifier performance, particularly in a class imbalance setting, before discussing variable importance in non-parametric models.

Readers familiar with the algorithms and the topic of supervised learning, can skip section 4.2 through 4.3.5

4.1 Classification

Classification in statistics refers to methods and models that seek to predict response variables that are categorical. For example, a flower petal's colour can assume several different colours such as red, blue, or white, making the petal's colour values qualitative. Predicting categorical values for a given observation is referred to as classifying that observation, as it assigns a given observation to a class or category (James et al., 2013).

Classification models usually generate two different types of prediction. First, a continuous valued prediction is made, usually in the form of a probability, in much the same way as a regression model. Then, a predicted class which comes in the form of a discrete category is created. The predicted class is assigned according to the probability of class membership (Kuhn and Johnson, 2013).

4.2 Supervised Learning

Supervised learning refers to learning methods that can map inputs based on example input-output pairs, meaning that the model is provided with examples of correctly labeled observations, and told to find the patterns leading to the correct label (James et al., 2013). Supervised learning is often considered the standard learning method, and distinguish it self from unsupervised learning where the output is not known a priori (Kuhn and Johnson, 2013). Supervised learning models have a mathematical structure can be described as prediction y_i is made from input X_i.

In supervised learning, models performance in learning is evaluated by measuring *training loss function* and *regularisation function* (Friedman, 2001; Chen et al., 2020). Jointly, these functions make up the *objective function*, defined in equation 4.1 , which is used to measure how well the model fit the training data. This function is optimized iteratively while constructing the model (Chen et al., 2020).

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{4.1}$$

Ω denotes the regularization term, and L denotes the training loss function. The training loss is a measure of how predictive our model is on the training data. A common choice of loss function is *mean squared error* or, for logistic regression, *logistic loss.* In this thesis we utilize the logistic loss function,

$$L(\theta) = \sum_i [y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i)\ln(1 + e^{-\hat{y}_i})] \tag{4.2}$$

where y_i is the true class label (litigation or no-litigation) and $\hat{y}_i$ represents the predicted class label, which can be derived from a linear model, exemplified in equation 4.3, or could be the logistic model discussed in section 4.3.1

$$\hat{y}_i = \sum_j (\theta_j x_{ij}) \tag{4.3}$$

The regularization term, $\Omega(\theta)$, is important as it controls the complexity of the model, which helps in avoiding model overfit. Overfit occurs when the model is trained to correspond too closely to a specific set of data, often the training data, and is unable to accurately capture unseen data (James et al., 2013). The concept is illustrated in figure 4.1.

The concept of overfitting and the concept of bias-variance trade off are closely related. In statistical modeling we seek to produce models that both have low bias and low variance. However, the increase in one is often associated with an increase in the other. Simple models might have low variance, as they tend to be less flexible and to a lesser extent

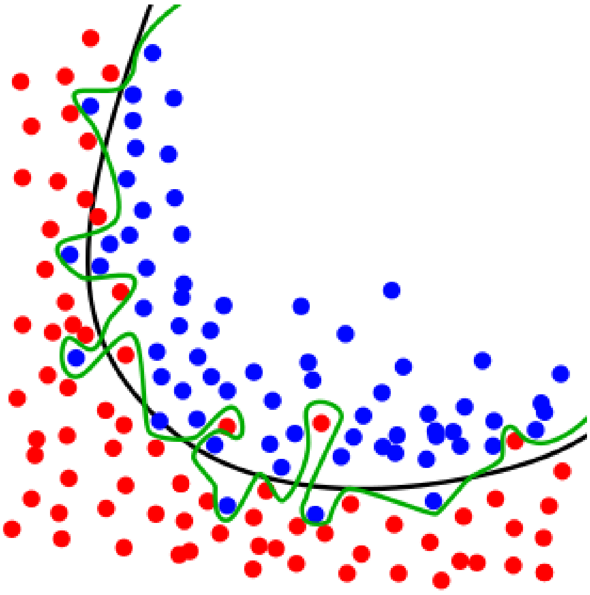


Figure 4.1: Overfit in a binary classification setting - The green line represents an overfitted model while the black line represent a regularized model. The blue and red dots represents two class labels (Chabacano, 2020)

react to changes in input, but would accordingly have high bias as a simple approximation leads to modeling error (James et al., 2013). The key to reducing prediction error is to find the best balance between variance and bias, and is referred to as the bias-variance trade off. The concept is illustrated in figure 4.2

Higher values of $\Omega(f)$ lead to more complex models. As such, keeping $\Omega(f)$ as low as possible is an objective under construction of predictive models. A general heuristic is that we seek to produce simple and predictive models (Chen et al., 2020).

All algorithms used in this thesis are supervised learning algorithms, and as such, adhere to the concepts outlined in this section.

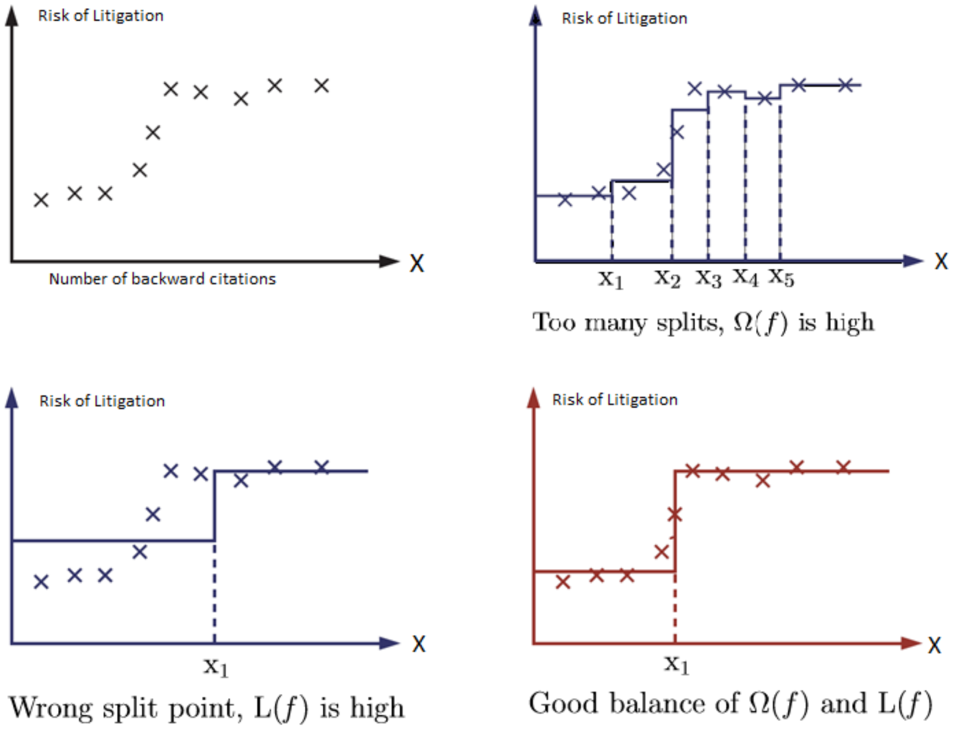


Figure 4.2: The Bias-Variance trade off - The Loss function and the Regularization term (Chen et al., 2020)

4.2.1 Training and Test Evaluation

In supervised learning, high variance due to overfitting can be hard to negate as models learn to fit the data patterns in training, regardless of the amount of regularization used. If the model is trained on a specific dataset and its structure, it will inherently achieve higher variance. If we were to measure the training model's performance based on its ability to produce results on a dataset that never changes, it would most likely not be a model that is capable to respond to new and unseen data (James et al., 2013; Kuhn and Johnson, 2013). There is a distinction between *training error* and *test error*. Training error relates to the performance of the model in training, and is used to evaluate and select the best fit in the model training process. The test error is used to evaluate the performance of the final fit on "realistic" and entirely unseen data (James et al., 2013). The measure of these rates is discussed in section 4.5.

Test Error and Final Model Evaluation

In order to evaluate the performance of a trained model and its final fit, a portion of the full dataset is split off into a test set. The performance of the final fit is measured as the test error - the error that results from using a learning method to predict a response on a new observation (James et al., 2013). In this thesis, the test set is randomly split in a 70/30 partition, where 70 % of the data is used in training.

Training Error and Cross Validation

A common solution in overcoming problems of overfitting in the training of the models is to create a third partition of data by *holding-out* a part of the training data, in order to evaluate the training performance. This is often referred to as the hold-out method or the validation set approach, and the dataset created is referred to as the validation set (James et al., 2013). By splitting the training data into two parts, the performance of the training can be evaluated on a dataset not previously seen by the model. Thus, the best fit and the best hyperparameter settings can be selected based on the performance on the validation set. The performance is measured as the training error. A drawback of this approach is that our already reduced dataset for training is further reduced, and as such not all data available is utilized in the training process (James et al., 2013).

As 30 % of the data is removed to a test set before training, it is often not desirable to split the remaining 70 % further into training and validation sets. Cross-validation offers a solution where the dataset is split into k partitions and used once as the validation set, and k - 1 times as training set. The number of iterations, or folds, are dictated by the number of partitions, as all partitions needs to be used one time. By using this method, not only is the validation set more varied and thereby decreasing variance, bias and risk of overfit, but also enables the use of all data available. These cross-validation methods are often referred to by the number of folds (partitions). Popular choices include 5-fold and 10-fold cross validation (Kuhn and Johnson, 2013; James et al., 2013).

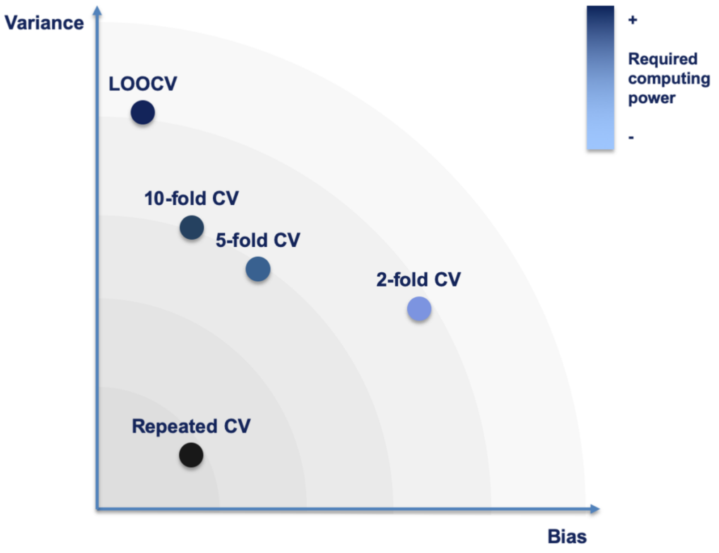


Figure 4.3: Validation methods relative to bias-variance trade-off (Feick, 2019)

While cross-validation reduces the variance of a model, there are better alternatives. Repeated cross-validation is a variant of the k-fold cross-validation method that repeats the cross-validation for a set number of times and than take the average (Kim, 2009). Kim (2009) finds that repeated cross-validation outperforms traditional cross-validation methods such as k-fold cross-validation, in addition to other alternatives such as the bootstrap (discussed in section 4.3.2). Repeated cross-validation reduces the variance further as compared to k-fold cross validation, while keeping the bias low. However, the repeated cross validation method requires much more computational power. Figure 4.4 shows different cross-validation methods compared by the bias-variance trade-off and computation requirements. Repeated cross-validation has less variance and less bias at the cost of heavier computation.

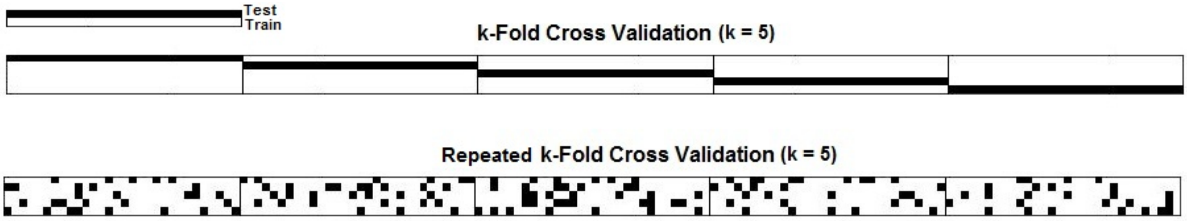

Test
Train
k-Fold Cross Validation (k = 5)
Repeated k-Fold Cross Validation (k = 5)

Figure 4.4: 5-fold cross-validation and repeated 5-fold cross-validation (Kapil, 2018)

The terms *training set*, *validation set*, and *test set* are often confused in machine learning theory (Ripley, 2007). Erroneously validation set is often referred to as test set, which can create confusion as they fulfill many of the same purposes, albeit at different stages of model building. The training set is used to train the model, the validation set is the set that is used to evaluate the training and tune hyper parameters, the test set is used to measure the performance of the final model in terms of accuracy/sensitivity/ROC et cetera (James et al., 2013; Kuhn and Johnson, 2013). In some cases there is no dedicated test set, such that the validation set becomes the test set (and the validation set is then correctly referred to as the test set). Where dedicated test set is lacking, cases cross-validation becomes much more important in order to ensure that the model does not overfit and both variance and bias are reduced.

In this thesis, all models use repeated 5-fold cross-validation with 5 repeats as the validation set. 5 folds and 5 repeats where deemed sufficient to reduce variance and bias while at the same time being more computationally efficient than a 10-fold version.

4.3 Classifiers

In this section we present the four classifiers used in our studies: Logistic Regression; Random Forest; XGBoost; and Support Vector Machines.

4.3.1 Logistic Regression

Logistic regression aims to describe the relationship between independent variables and a dependent variable (Kleinbaum and Klein, 2010). The logistic regression classifier, often referred to as *logit* in cases with a binary dependent variable, is a machine learning algorithm that uses regression to calculate the probability of classes membership and assign class accordingly. Kleinbaum and Klein (2010) define the logistic function $f(z)$, which describes the mathematical form of the basis of logistic model, as

$$f(z) = \frac{1}{1 + e^{-}z} \tag{4.4}$$

As shown in equation 4.4, the range of $f(z)$ is between 0 and 1, regardless of the value z . Kleinbaum and Klein (2010) cite the range property as one of the primary reasons for the logistic regression models popularity, as it is easy to understand and insensitive to scale. To users which are gauging the risk associated with different outcomes, such as epidemiology, credit fraud, or insurance, the range of $f(z)$ is particularly useful.

Kleinbaum and Klein (2010) further state that some of the logistic regression's popularity stems from shape of the logistic function and its interpretations. The S-shape of the logistic function communicates the combined effect of several predictors on the outcome, and can be very useful in communicating the result of the model and the understanding of the underlying risk.

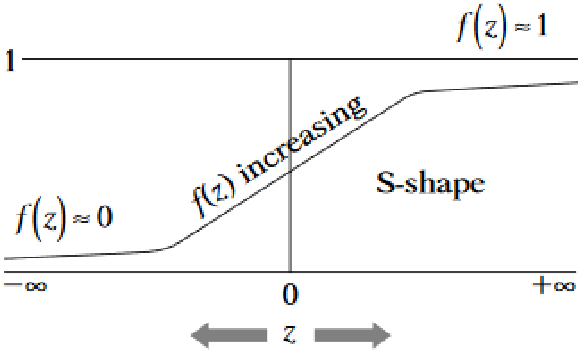


Figure 4.5: The shape of the $f(z)$ logistic function (Kleinbaum and Klein, 2010)

In order to obtain the logistic model from the logistic function, z is written as the linear sum of $\beta_0 + \beta_p \times X_p$, where X_p are the independent variables and β_0 and β_p are constant terms representing unknown parameters (Kleinbaum and Klein, 2010). z can therefore be written as

$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p \tag{4.5}$$

If we substitute the expression in equation 4.5 for z in the logistic function, we get

$$P(\mathbf{X}) = \frac{1}{1 + e^{-(\beta_0 + \sum \beta_p X_p)}} \tag{4.6}$$

Where $P(\mathbf{X})$ is shorthand for the probability $Pr(Y = 1|X_p)$

The logistic function is also commonly written as

$$p(X) = \frac{e^{\beta_0 + \beta_p X_p}}{1 + e^{\beta_0 + \beta_p X_p}} \tag{4.7}$$

The Log-Odds

In many contexts the odds can be a preferred as a method of communicating the probability of a given outcome, and has applications in interpreting the coefficients yields be the logistic regression model (James et al., 2013). The logistic model in equation 4.7 can be rewritten as

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_p X_p} \tag{4.8}$$

The left side of equation 4.8 is referred to as the odds, and can assume any value between 0 and ∞. Low values of odds close to 0 indicates very low probabilities of a positive, while higher values towards ∞ indicates higher probabilities of a positive (James et al., 2013). In the context of the subject of our paper; on average 1 in 5 patents with an odds of $\frac{1}{4}$ will go into litigation, as $p(X) = 0.2$ implies and odds of $\frac{0.2}{1-0.2} = \frac{1}{4}$.

By taking the logarithm of both sides of equation 4.8 we get the log-odds (James et al., 2013), written as:

$$log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_p X_p \tag{4.9}$$

The log-odds is useful in understanding the relationship between independent variables and likelihood of class membership. Increasing an independent variables (X) with one unit is associated with a change in log-odds by β_1. As the relationship between the likelihood of class membership ($p(x)$) and the independent variables (X) is not a straight line, a one-unit increase in the β_1 parameter does not correspond to a change in $p(X)$ (James et al., 2013). However, a positive β_1 parameter will be associated with an increase in $p(X)$.

To illustrate the intuition behind the relationship between the coefficients and the likelihood of class membership, we can consider a case where we have patent with a single independent variable "number of backwards citations". If the estimated coefficient is 0.0062, a one-unit increase in "number of backwards citations" is associated with an increase in the log odds of litigation by 0.0062.

Maximum Likelihood - Estimating the Regression Coefficients

The parameters β_0 and β_1 in equation 4.7 are unknown, and have to be estimated based on the available training data. The maximum likelihood method is often preferred method of estimation (James et al., 2013). The intuition behind the maximum likelihood method is as follows: We seek estimates for β_0 and β_1 such that the predicted probability $\hat{p}(x_i)$ of a positive for each observation, utilizing the logistic model from 4.6, matches as closely as possible the observed class of any observation (James et al., 2013). Or in the context of the subject of this paper, we try to find β_0 and β_1 such that plugging the estimates into the model for *p(X)* given in 4.7, gives us a number that is close to one for all patents that litigated, and a number as close to zero for all patents who did not. This intuition can be formalized in the *likelihood function.*

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)) \tag{4.10}$$

The estimates $\hat{\beta_0}$ and $\hat{\beta_1}$ are chosen in a manner that maximizes the likelihood function shown in equation 4.10

4.3.2 Decision Tree Ensembles

Decision trees have in recent years become increasingly popular as they are both intuitive to understand, can be displayed graphically, can handle categorical independent variables without the need to create dummies, and are proved to produce good results (James et al., 2013; Khalilia et al., 2011; Kuhn and Johnson, 2013).

Random Forest and *Gradient Boosted Trees* are both Decision Tree ensembles in that they combine the results of a series of separate decision trees to make their predictions (Khalilia et al., 2011; Chen and Guestrin, 2016; Chen et al., 2020). Both utilize a variant of decision trees that produce decision scores, rather than just true/false labels, referred to as Classification and Regression Trees (CARTs) (Loh, 2011). A simple CART is shown in figure 4.6.

Patents are classified into different leaves and are assigned a prediction score corresponding to the score in the leaf. The score generated by CARTs enables richer interpretation of results, as compared to ordinary decision trees.

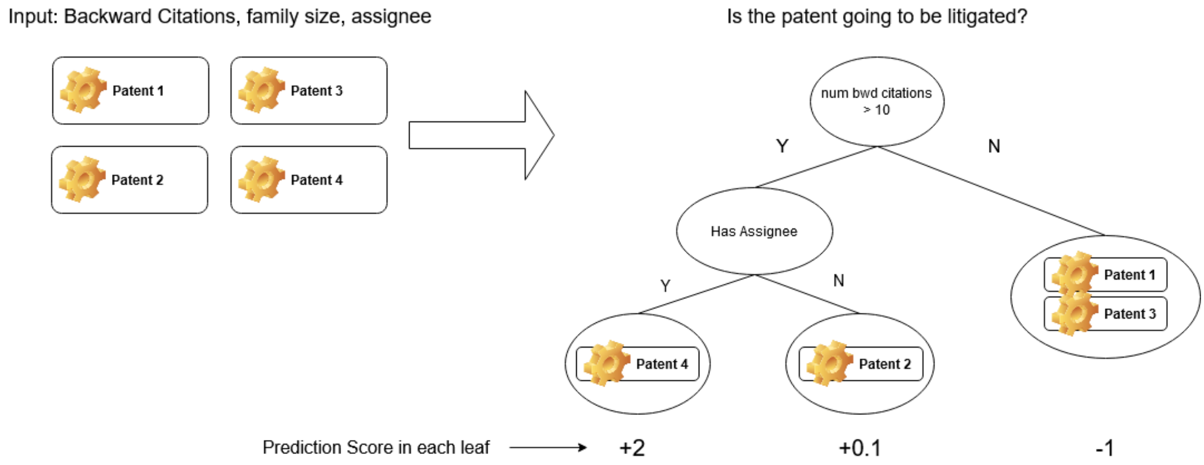

Input: Backward Citations, family size, assignee
Patent 1
Patent 3
Patent 2
Patent 4
Is the patent going to be litigated?
num bwd citations > 10
Y
N
Has Assignee
Y
N
Patent 4
Patent 2
Patent 1
Patent 3
Prediction Score in each leaf
+2
+0.1
-1

Figure 4.6: A simple CART (Chen et al., 2020)

A single CARTs prediction score is not good enough by it-self. In tree ensembles multiple trees are created and prediction scores of leaves are summarised to create the prediction. The intuition of ensembles, where the score of corresponding leaves in each tree is summed to a single predicted score, is illustrated in figure 4.7.

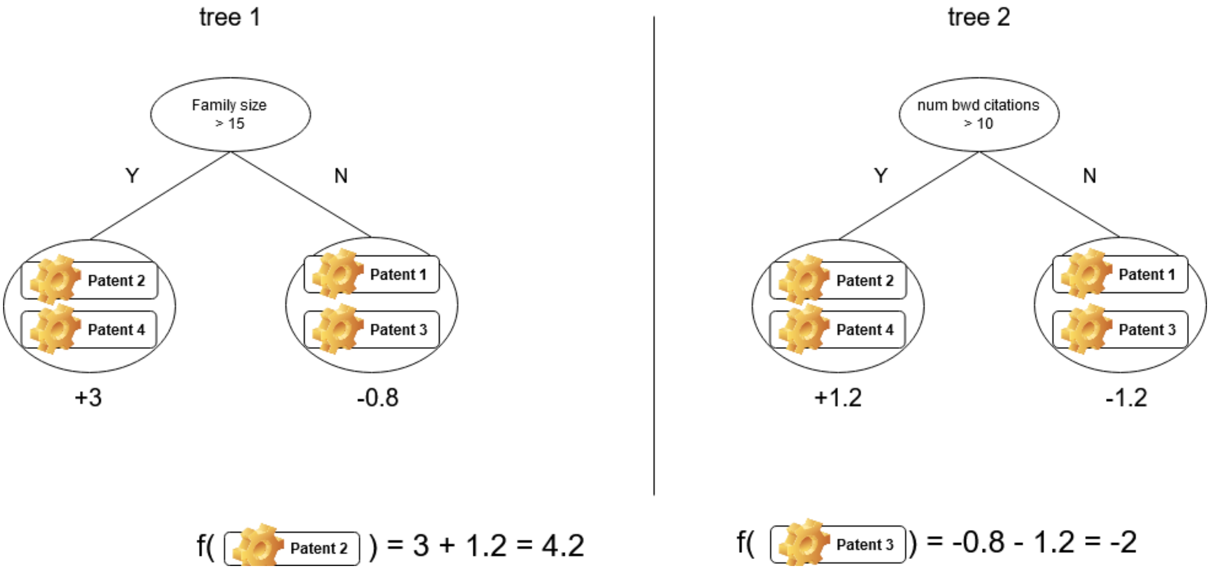


Figure 4.7: An example of a tree ensemble (Chen et al., 2020)

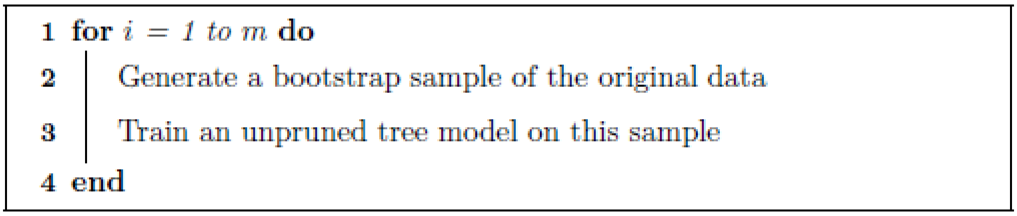
Each model in the ensemble is used to generate a prediction for a new sample. Each tree will cast a vote for some input x, and the output of the classifier is determined by total score. The input x with the most votes is selected as the output of the classifier (Khalilia et al., 2011). As each learner is created, the algorithm evaluates the performance by the objective function defined in equation 4.1

Both Random Forest and XGBoost use the decision tree ensemble algorithm. The way in which they differ is how they are trained. Random Forest uses *bagging* to create learners, while XGBoost uses *boosting*.

Bagging and the Bootstrap

Bootstrap aggregation, or *Bagging*, is a variance reduction method in a statistical learning model by aggregating over a set of bootstrapped training sets (James et al., 2013). A bootstrap (Efron and Tibshirani, 1986) sample is sample of data with *replacement*, meaning that a single observation can be re-used in creating a new sample of data (Kuhn and Johnson, 2013). The bootstrap samples known observations in order to calculate measures of accuracy such as bias and variance. The basic intuition of bootstrapping is that the distribution of a population can be inferred by the distribution of a resample (James et al., 2013).

In bagging, originally proposed by Breiman (1996), multiple bootstraps from a single training set are created. The model is then trained on all bootstraps and the predictions are averaged (James et al., 2013; Kuhn and Johnson, 2013). The advantage in introducing bagging to a model is that it reduces variance (as it is trained on many different data samples) without increasing bias. This is particularly useful in tree-models, as they tend to suffer from high variance and unstable results (Kuhn and Johnson, 2013). Bagging allows for parallel processing as all trees are created at the same time (Breiman, 1996).



```
for $i$ = $1$ to $m$ do
    Generate a bootstrap sample of the original data
    Train an unpruned tree model on this sample
end
```

Figure 4.8: The Bagging algorithm (Kuhn and Johnson, 2013)

The bootstrap if often considered as a viable option to cross-validation, and solves many of the same problems (Kim, 2009). None-the-less we have chosen to run a repeated cross-validation on all models, including those that utilize bootstrap, in order to give all model the same format of input. However, this is likely redundant in our case, and can be considered excluded in future work or application in order to improve run-time.

Boosting

In *boosting*, like bagging, a large number of decision trees are combined to produce predictions. However, in contrast to bagging, where a separate decision tree is fitted to each bootstrap set independently and then combined, boosted trees are grown sequentially with weak learners. Moreover, boosting does not involve the use of bootstrap sampling, instead each tree is fit to a modified version of the original dataset (James et al., 2013).

Unlike bagging, where a single decision tree is fit to the data, which can lead to overfitting, a boosting approach will learn slowly as trees are fitted sequentially. Instead of fitting the model to the outcome Y, a boosted model will be fitted based on the current residuals and slowly build with smaller trees (James et al., 2013; Kuhn and Johnson, 2013).

Trees are in particular suited for boosting as they can easily be made into weak learners by reducing the number of terminal nodes (tree depth); they are easily added together; and they can be generated very quickly (Kuhn and Johnson, 2013). However, computation time in boosting methods are often greater than for methods using bagging (such as Random Forest).

4.3.3 Random Forest

The Random Forest algorithm (Breiman, 2001), shown in figure 4.9, is a decision tree ensemble learner that uses bagging to train CARTs, and is thus a variant of a bagging model (Khalilia et al., 2011). However, in contrast to a regular bagging model, Random Forest introduces an element of randomness that decorrelate the trees. When building a tree in Random Forest, each time a split is to be made, a random sample of predictors are considered as split candidates as a subset of the full set of predictors.

The rationale of introducing randomness to the available split candidates is related to the tendency of trees being correlated due to strong predictors in the top levels of the trees. Consider a case where there is one strong predictor and several moderate predictors in the set. In this case almost all trees will use the single strong predictor in the top split, thus making the trees very similar in structure. This is a weakness in bagging. Averaging over a set of correlated trees does not reduce variance as much as averaging over decorrelated trees. Random Forest overcomes this problem by forcing the trees to only consider a smaller subset of the available predictors (James et al., 2013). Random Forest has shown it self to preform well on a multitude of different classification problems (Khalilia et al., 2011; Probst et al., 2019).

The Random Forest Algorithm

```
1. $ntree$ = number of trees to be generated
2. $N$ = number of samples in the dataset
3. for $t = 1$ to $ntree$ do
    (a) Generate bootstrap sample $Z$ of size $N$ from the original data - with replacement
    (b) for each bootstrap $Z$ grow a classification tree
    (c) for $i = 1$ to $number\boldsymbol{of}nodes$ do
        i. randomly sample $mtry$ predictors from $M$ predictors
        ii. choose best split among the sampled predictors
    (d) end for
    (e) $Y_t(x)$ = class prediction of the $t$th tree
4. end for
5. $Y_{rf}(x)$ = majority voting $\{y_t(x)\}^{ntree}{}_1$
   (Final predicted class is majority voting over all trees in RF)
```

Figure 4.9: RF algorithm (Khalilia et al., 2011)

Hyperparameters and the Splitting Criterion

In selecting the split, Random Forest utilizes the Gini Index as a criterion. The Gini index is a measure of total variance across K classes and is defined as (Khalilia et al., 2011; James et al., 2013):

$$G = \sum_{K=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{4.11}$$

Random Forest does not consider all variables at each split, but relies on a subset. Therefore, on average $(p - m)/p$ of the splits will not consider the strongest variable, thus *decorrelating* the trees, making them less variable and more reliable.

If we assess point i from the algorithm in figure 4.9 , we can see that in each split decision the sample of variables used is randomly chosen. The variable sample selection in each split is made from a random subset m of variables derived from the full set of variables p. Typically, a fresh sample of m variables at each split is chosen as $m \approx \sqrt{p}$ - such that the number of variables considered at each split is approximately equal to the square root of the total number of predictors (James et al., 2013).

$\hat{p}_{mk}$ represents the proportion of training observations in the mth region that are from the kth class. The Gini Index takes values from 0 to 1, where 0 is obtained when all classes in the node has the same label (Khalilia et al., 2011). Due to this property, the Gini Index is often referred to as an measure of node purity (James et al., 2013). The decision of the criterion will be based on the lowest computed Gini index amongst the predictor.

The splitting rule is typically not considered a hyperparameter in it self, as it characterizes the Random forest Algorithm. However, it can in a large sense be considered a categorical hyprerparameter where the standard measure of Gini impurity can be substituted with other splitting criteria (Probst et al., 2019). For example, in order to increase computational efficiency, Geurts et al. (2006) propose a randomized splitting rule.

The *nodesize* hyperparameter specifies the minimum number of observation in a terminal node. A lower set node size leads to trees with larger depth, meaning that more splits are performed until that node ends (Probst et al., 2019). A minimum node size of 1 is standard for classification in many learner libraries (Probst et al., 2019).

4.3.4 Gradient Boosted Trees

Gradient Boosted Trees (Friedman, 2001) is, as Random Forest, an ensemble method. Where Random Forest uses bagging, Gradient Boosted Trees uses boosting. In gradient boosting we seek to find, given a loss function (e.g., logistic loss) and a weak learner (can be shallow CARTs), the additive model that minimizes the loss function. The model typically starts with the best guess of the response, the gradient is then calculated and a model is fit to the residual such that the loss function is minimized. The current model is fit to the previous model. This process is repeated for a set number of iterations (Kuhn and Johnson, 2013; Friedman, 2001).

The main difference between gradient boosting and Random Forest is the use of weak learners in combination with the calculation of the gradient for each iteration.

XGBoost

Extreme Gradient Boosting, or XGBoost for short, is an implementation of gradient boosting trees which is designed to increase speed and performance. XGBoost has in recent years shown to perform well in a number of settings and also in competitions hosted by machine learning competition site Kaggle (Chen and Guestrin, 2016). Chen and Guestrin (2016) state that one of the key advantages of XGBoost is its scalability in all scenarios and superior run times. This scalability is due to a series of key innovations in XGBoost over traditional boosting methods, which includes weighted quantile sketch procedure enables instance weights in tree learning and novel tree learning algorithms for handling sparse data. Faster model exploration is enabled by parallel and distributed computing, which particularly beneficial when single-threaded statistical program R is used. A key advantage is XGBoost's exploitation of out-of-core computation, processing of data that are too large to fit into the computer's RAM at once, which enables the processing of millions of observations (Chen and Guestrin, 2016). All these techniques and innovations on gradient boosting are in XGBoost combined into an end-to-end system that scales to large data with the least amount of cluster resources (Chen and Guestrin, 2016).

While regular gradient boosting has two tuning parameters: tree depth and number of iterations (Kuhn and Johnson, 2013), XGboost offers greater flexibility through more tunable hyperparameters. These include: regularization through the shrinkage parameter *eta*; a choice on the depth of each tree (how many variables each tree will split on growing depth-wise); a fraction for bagging; L2 regularization term *lambda* that resembles ridge regression that punishes the squared size of variable weights; *gamma* denotes the minimum loss required for the split of a tree. The column subsampling fraction makes the fit of the model behave more like a random forest classification, enabling stochastic gradient boosting, where a share of predictors are chosen randomly for each iteration.

4.3.5 Support Vector Machine

Support Vector Machines (SVM) are a group of highly flexible and powerful modeling techniques originally developed as classification models, that can also be extended to regression. The approach was developed in the 1990s and according to James et al. (2013) has grown in popularity since. The objective of Support Vector Machine algorithm is to find a hyperplane in a multidimensional space, where the number of dimensions is the equal to the number of features in the dataset. The are many variations of hyperplanes that can be chosen for separation of the classes. The objective is to find the hyperplane that has the maximum margin, i.e. the "longest" distance between the observations from the dataset that the hyperplane seeks to separate (Figure 4.10). This facilitates for future observations to be classified with more confidence.

The points that are closest to the separating hyperplane are called *support vectors.* These points influence the position of the hyperplane, and addition or deletion of these points will change the position of the hyperplane. For some classification problems, separation by hyperplane is not possible. It is however possible to allow slack in the margins of the hyperplane. That decision allows some observations to be located on the "wrong" side of the margin (James et al., 2013).

Finding the maximal margin hyperplane, is an optimization problem in which the width of the margin M is maximized, given by non-negative constraints of the slack variable ϵ_i and the total amount of allowed slack cost C. Slack cost C can be tuned in order to find the optimal amount of cost allowed in the model (James et al., 2013).

$$maximize\, M,\; subject\, to\, \epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \tag{4.12}$$

The classifier for maximal margin classifies test observation x*, where $\beta_0 + \beta_1 ... \beta_p$ are the coefficients of the maximal margin hyperplane.

$$f(X) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + ... + \beta_p x_p^* \tag{4.13}$$

The decision boundaries can also be non-linear, and SVM methods can be used both on two- or multiple-class classification problems. By using a non-linear kernel that transforms the data with complex calculations in order to let the SVM to find optimal hyperplanes. In that case the classifier takes the form as shown in figure 4.14. Simply put, kernels are functions that convert a problem that is not separable into a separable problem. The most used kernel is Radial Basis Function (4.15), due to the fact that it is general purpose and no assumptions about the data must be made a priori.

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i) \tag{4.14}$$

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2) \qquad (4.15)$$

Since we do not know whether our dataset is linearly separable or not, both methods of SVM that are described will be implemented and analyzed.

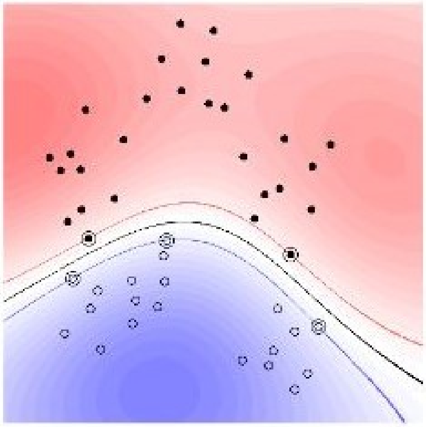


Figure 4.10: Example of a SVC determined by a radial basis function kernel. Black and white point are two classes, the middle line is the decision hyperplane and the lines on each side of the hyperplane are the margins. Support Vectors found by the algorithm are examples that are critical for the classification (marked with extra circles) (Schölkopf et al., 1996).

4.4 Class Imbalance

In modeling discrete classes, the relative frequencies, that is the frequency of classes which occurs in the data on which we are modeling, can have a substantial impact on the effectiveness and performance of the model (Kuhn and Johnson, 2013). An imbalance of classes occurs when one or more classes display very low proportions in the training set, as compared to other classes. In the case of a binary classification problem, class imbalance occurs when one of two classes has a significantly larger proportion than the other in the training set. Modeling classification problems on imbalanced data is often referred to as anomaly detection, as positives are considered "out of the normal" when the ratio of negatives and positives are severely imbalanced (Chandola et al., 2009)

In classification, imbalance creates several problems that can affect a model's performance. When the positive class are infrequently present, they are most likely predicted as rare occurrences, undiscovered, ignored, assumed as noise or outliers, which can result in more frequent misclassifications of the positive class (Ali et al., 2015).

4.4.1 Sampling

A widely used technique to remedy the effects of class imbalance on classifiers performance is to *sample* the data to create a different distribution of labels before a model is trained (Kuhn and Johnson, 2013). The two general forms of sampling approaches are over- and under-sampling.

In under-sampling, the number of observations in each class is dictated by the number of observations in the minority class (Kuhn and Johnson, 2013). For example, in our dataset there are 25,800 positive observations. If we under-sample our dataset there would be 25,800 negatives and 25,800 positives. An advantage of under-sampling is that it does not introduce any synthetic data, and as such yields less bias.

In over-sampling, the number of minority class observations are "up-sampled" such that the number of observations in the minority class is dictated by the number of majority observations. This method introduces repeated data to achieve more observations. Over-sampling can be preferred when the total number of observations are low, or when the number of minority class observations is too low for under-sampling.

Other popular choices are Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) and Random Over-Sampling Examples (ROSE) (Menardi and Torelli, 2014). Both have been tested on our data and found not to offer any improvement on under-sampling. Therefore, these methods are not discussed at length in this thesis. Interested readers are refereed to Chawla et al. (2002) and Menardi and Torelli (2014).

4.5 Model Evaluation

4.5.1 Class Specific Probabilities

As mentioned, classification models produce both discrete and continuous predictions. In terms of assessing a models performance, the focus is often on the discrete predictions, as they are connected more directly to the end result of an observation's classification. However, it can be useful to assess the continuous predictions, as they can give us information about the model's confidence of the predicted classification (Kuhn and

Johnson, 2013). For example, with a threshold of 0.5 probability for class allocation, an observation with 0.51 probability of class membership is given the same class as an observation of 0.99 probability of class membership. Although being classified in the same class, one is more confident that the latter observation is correctly classified than the former.

4.5.2 Evaluation Metrics

Confusion Matrix and Overall Accuracy

Confusion matrix is a tool commonly used to describe the performance of classification models (Kuhn and Johnson, 2013). A confusion matrix is made up of a simple cross-tabulation of the predicted and the observed classes used in modeling. Diagonal rows denote correctly predicted classes, while off-diagonal rows denote the number of errors for each case. A generic confusion matrix is presented in Table 4.1.

	Observed	
Predicted	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Table 4.1: A generic confusion matrix

The number of correctly predicted classes associated with the positive outcome, known as true positives (TP, top-left corner). The number of correctly predicted classes associated with the negative outcome, known as true negatives (TN, bottom-right corner). The total amount of correctly predicted classes is the sum of true positives and true negatives. The number of incorrectly predicted negative and positive classes, known as false negatives (FN, bottom-left) and false positives (FP, top-right). The total number of incorrect predictions is the sum of false positives and false negatives. Thus, the confusion matrix offers a simple overview of the performance and a models ability to correctly predict classes.

Overall accuracy, also known as the error rate, is a simple measure of the relationship between predicted and observed classes. Overall accuracy is defined as the total number of correct predictions divided by the total number of observations (Kuhn and Johnson, 2013), written as

$$Overall\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.16}$$

Overall accuracy is both easy to understand and compute, and there is some value in understanding the overal accuracy of a model. However, there are issues in relying on overall accuracy as a performance metric. Firstly, overall accuracy makes no distinction between what type of errors are being made in the prediction (Kuhn and Johnson, 2013). In different applications, the type of error is given importance according to the consequences of the given error type. Secondly, overall accuracy is known to be an inappropriate performance metric for rare event classification problems, such as medical diagnosis, fraud-detection, click rate prediction, or patent litigation (Koyejo et al., 2014b). In such cases, the problem arises from the fact that high overall accuracy can be achieved by predicting all classes as non-events (negatives) (Kuhn and Johnson, 2013), thereby create a "null-classifier" model. In our data, where 0.8 % of all patents are litigated, the null-classifier rate is 0.92. Thus, high accuracy can be achieved without adding any new information. Therefore, tuning a model and evaluating its performance by overall accuracy is discouraged, especially in cases of rare event classification problems. Koyejo et al. (2014a) states that tuning rare event classification models with overall accuracy are well known to be inappropriate, and alternative metrics better tuned to imbalanced classification are preferred.

Sensitivity and Specificity

Sensitivity and Specificity are metrics that address the issue of a model's error rate. Sensitivity, often referred to as the true positive rate (TPR) or recall, measures the accuracy of the positive population. Sensitivity is the number of positive samples that is predicted to positive over the number of actual positive samples, or

$$Sensitivity = \frac{\#\text{ positive samples and predicted to be positive}}{\#\text{ positive samples}} = \frac{\text{TP}}{\text{TP + FN}} \tag{4.17}$$

Specificity, often referred to as the true negative rate (TNR), measures the accuracy of the negative population. Where specificity is the number of negative samples predicted to be negative over the number of actual negative samples, or

$$Specificity = \frac{\#\text{ negative samples and predicted to be negative}}{\#\text{ negative samples}} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4.18}$$

Assuming a fixed level of accuracy for a model, there is typically a trade-off associated with using sensitivity and specificity as performance metrics, as an increase in sensitivity is likely to incur a loss in specificity (Kuhn and Johnson, 2013). This trade-off is perhaps more obvious if we consider that specificity can be written as one minus sensitivity and vice versa. These potential trade-offs may be appropriate when there are different importance and penalties associated between the type of errors (Kuhn and Johnson, 2013). For instance, in cases where false negatives are critically important such as cancer detection, there is usually focus on sensitivity, as one would be willing to accept some cases where patients where falsely diagnosed with cancer in order to minimize the number of patients with cancer that is not detected. Therefore, both specificity and sensitivity are most useful as performance metrics when the importance of which error types are made is not equal.

An often overlooked aspect of sensitivity and specificity is that they are conditional measures (Kuhn and Johnson, 2013). In terms of measuring performance of a model, sensitivity and specificity are useful. However, for an end-user of a model, these conditional measures are not very useful. For instance, an end-user is typically not interested in conditional queries such as "given that the patent is not going to litigation, the model has an accuracy of 95 % ?", but rather unconditional queries such as "what is the probability of this patent going to litigation?".

4.5.3 Problems in Using Accuracy Based Metrics

Although accuracy based metrics are the most commonly reported model evaluation metrics (Akosa, 2017), there are two main issues in utilizing these metrics in model evaluation; fit for purpose problems; and the effect of class imbalance. Both issues are important in deciding on model evaluation metrics.

Fit for Purpose - Accuracy Based Metrics and Limitations in Application

Although accuracy remains important, it has limited in usefulness in many real world applications, where the overall accuracy is not the end goal (Kuhn and Johnson, 2013). If we want to quantify the consequences of correct and incorrect predictions, other metrics that takes these consequences into consideration should be considered. For example, in our patent litigation problem a false negative prediction, that is a patent predicted to not go into litigation actually does go into litigation, can have much greater consequences on the predicted cost of that patent. Any model should consider the extra cost associated with applying false negative predictions, and should be tuned thereafter. Conversely, from an insurers perspective false negatives can be very costly, but also harmful to a potential patent litigation insurance customer, as it would drive the cost of insurance premiums up if a model is tuned to minimize false negatives. Therefore, the purpose of which a model is fit carry great importance.

Accuracy Based Metrics and Class Imbalance

In section 4.4, we discussed some of the problems and remedies in cases where class imbalance occurs. The techniques discussed mostly deal with model performance during parameter tuning, and thus the performance measure of the training subset. However, the best performing model is not chosen on the training subset, but rather on the test subset (Akosa, 2017). The distribution of the testing data may differ from that of the training data, and therefore the true misclassification cost may be unknown at the learning time. Moreover, the testing data needs to reflect the nature of the real data in order to produce honest estimates of future events (Akosa, 2017).

To illustrate the problem in using accuracy based metrics, consider a model which is built to detect credit card fraud. Assume that the data has 9,800 negative (non-event) cases and 100 positive (event) fraudulent cases. If the aim is to accurately classify fraudulent events and our classifier is built to optimize on accuracy, the classifier would predict all events as negative. The model would then achieve an accuracy of 99 %. However, the classifier does not offer any insights and would incorrectly classify all positive events as negative. Now consider a classifier that correctly predicts 60 of the cases to be fraudulent (positive) and incorrectly classifies 40 cases to be fraudulent giving the following confusion matrix:

	Obsereved	
Predicted	Positive	Negative
Positive	60	100
Negative	40	9,800

This classifier has an accuracy of 98.6 %. Even though this classifier has yielded correctly classified fraudulent cases, it has a lower accuracy score than the first classifier with zero predictive power. This phenomenon is referred to as the accuracy paradox, and highlights a problem in using accuracy based metrics in cases with class imbalance (Akosa, 2017). Provost and Fawcett (1999) write that the tacit assumption of constant and relatively balanced class distribution in using accuracy as a classification metric, makes it rarely applicable in real world situations. The utility of accuracy diminishes as difference in class proportions increases.

Utilizing class probabilities can potentially offer more information about model predictions than simple class values (Kuhn and Johnson, 2013). In the following sections we will discuss alternatives to accuracy based metrics that utilize class probabilities.

4.5.4 Receiver Operating Characteristic (ROC) Curves

Receiver Operating Characteristic (ROC) curve is a technique for visualizing and selecting classifiers based on their performance (Fawcett, 2006). Its origins and name derives from signal detection theory, and have long been used to depict the trade-off between hit rates and false alarm rates of classifiers. Spackman (1989) was one of the earliest adopters of ROC curves in classification, and demonstrated its utility in evaluating and compering algorithms. The use of ROC curves has gained popularity in the machine learning community, as it alleviates some of the problems of using accuracy-based metrics, as discussed in chapter 4.4. Moreover, ROC curves display properties that make them especially useful in domains where class distribution proportions are imbalanced, and or there are unequal classification error cost. These properties have gained increased importance as research into areas with cost-sensitive learning and learning on problems with imbalanced classes have become more commonplace (Fawcett, 2006). In this paper, ROC curves are only considered in a two-class setting in order to keep the paper concise and related to the problem space.

One of the more attractive properties of ROC is that it is insensitive to differences and changes in class distribution, meaning that a change in proportions in the test set does not affect the ROC curves (Fawcett, 2006).

The ROC Space

From section 4.5.2 we recall that the rate of which a model is able to predict positives are known as the True Positive Rate or Sensitivity and is defined by formula **??**. The rate of which positives are falsely predicted is known as false positive rate (or false alarm rate) and is given by the equation:

$$\text{False Positive Rate} = \frac{\text{Negatives incorrectly classified}}{\text{Total negatives}} \tag{4.19}$$

The ROC graphs are two-dimensiona, where the true positive rate is plotted on the Y-axis, and the false positive rate is plotted on the X-axis. A ROC graph depicts the relative trade-off between true positives (benefits) and false positives (costs) (Fawcett, 2006). In figure 4.11, a ROC graph with five classifiers labeled A though E from Fawcett (2006) is shown.

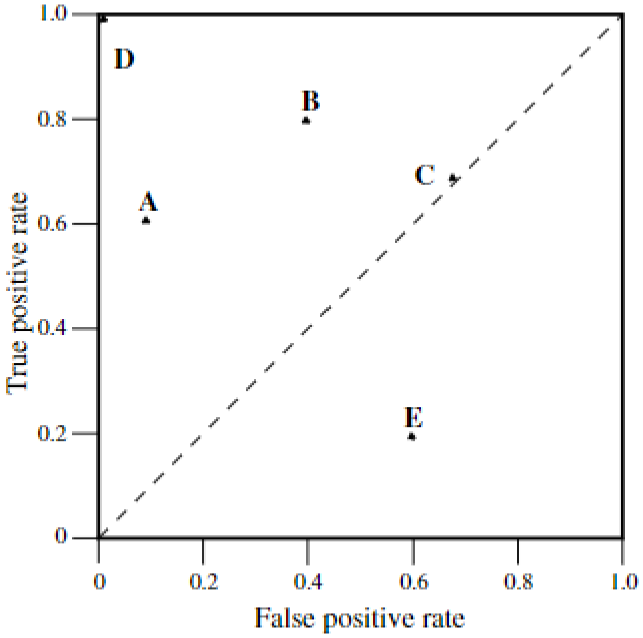


Figure 4.11: A Basic ROC graph displaying five discrete classifiers (Fawcett, 2006)

From section 4.1 we recall that a discrete classifiers is on the output only a class label, that is for example Litigation/nolitigation or fraud/nofraud. Each discrete classifier produces a false positive rate and a true positive rate which corresponds to a point in the ROC graph. The lower left point and the upper right point represents two different extremes of strategies. The lower left (0,0) represents a strategy where one refuses to make any positive predictions. Any classifier utilizing this strategy would make no false positive errors, but at the same time achieved no true positives. The upper right represents the inverse of this strategy, strictly issuing positives predictions, thus achieving maximum true positives at the cost of maximum false positives (Fawcett, 2006). The point (0,1), that is the top left corner, represent perfect classification with a true positive rate of 1 and a false positive rate of 0. In figure 4.11, the classifier D represents a perfect classifier. Informally, one point in the ROC space is better than another if it is north-west of the other, meaning that it has an higher TP rate, lower FP rate or both. Classifiers appearing near the X-axis on the left side of the space, can be described as "conservative", where any positive prediction is done on strong evidence (minimizing false positives). Conversely, any classifiers inhabiting the upper-right side of the space can be considered "liberal", as they require less certainty before making a positive prediction (Fawcett, 2006). Assessing figure 4.11, we can see that classifier A can be considered to be more conservative than B.

The diagonal dotted line $y = X$ represents a strategy of randomly guessing a class (Fawcett, 2006). For example, a classifier, given balanced class proportions, to randomly guess positives 50 % of the time, one would expect it to guess half the positives and half the negatives correctly; yielding the point (0.5 , 0.5) in the ROC space. If a classifier guesses positive 70 % of the time, it can be expected to give 70 % correct positives, but the false positive rate would also increase to 70%; yielding a (0.7 , 0.7) point on the curve. From figure 4.11, we can see that point C corresponds to a random guessing strategy. Thus, in order to "get away" from the $y = x$ line and into the top-left triangle, a classifier needs to employ some information in the data. Conversily, any point below the $y = x$ line would be performing worse than random guessing (Fawcett, 2006). However, studies by Flach and Wu (2005) shows that a classifier placed below the $y = x$ line is not necessarily useless, but rather is applying the information incorrectly. For readers interested the implications and solutions to classifiers performing under the $y = x$ curve, can refer to *Repairing concavities in ROC curves* (Flach and Wu, 2005).

Curves in the ROC space

Generally speaking there are two different outputs different classifiers produce. Discrete classifiers, such as decision trees and rule sets, produce a class decision on each instance (Fawcett, 2006). Therefore, when applied to a test set, such classifiers yields a single confusion matrix, that corresponds to a single ROC point. A threshold is applied to the classifier and each prediction is either above or below the threshold, that determines which class the instance is placed into. Each threshold value represents a point in the ROC space (Fawcett, 2006).

As high thresholds lead to less positive predictions, any predictions made under high thresholds would be considered "conservative". When the thresholds are lowered, the predictions move into the more "liberal" areas of the ROC space. In terms of movement in the space, a reduced threshold results in points moving up and right into the ROC space. Conversely, where the class distributions to be perfectly balanced, the classifier should perform best with a threshold of 0.5 (Fawcett, 2006). The curves themselves are created to be extrapolating between the points generated under different thresholds.

The Area Under the Curve (AUC)

As an ROC curve is a two-dimensional depiction of classifier performance and is most useful when visualizing the performance of a classifier as the decision threshold varies. Any point in the curve is a possible operating point for the classifier, and could be evaluated using accuracy measures (Bradley, 1997). In order to evaluate the entire curve, and achieve a singular scalar value to assess model performance, the Area Under the Curve (AUC) method is often preferred (Fawcett, 2006; Bradley, 1997). AUC is a portion of the area of the unit square and is therefore always between 0 and 1.0. AS mentioned, random guessing produces a diagonal line between (0,0) and (1,0). This diagonal line yields an AUC of 0.5, and therefore serves as a benchmark of any realistic classifier (Fawcett, 2006). Any AUC under 0.5 indicates that the classifier is either useless or applied incorrectly (Flach and Wu, 2005; Fawcett, 2006). An AUC score of 1.0 indicates a perfect classifier model that is able to predict all classes correctly over all thresholds. Thus, in measuring a classifier's performance on AUC we seek to achieve as high as possible AUC. Figure 4.12 shows a ROC curve of a generic logistic model, where the AUC is calculated as the

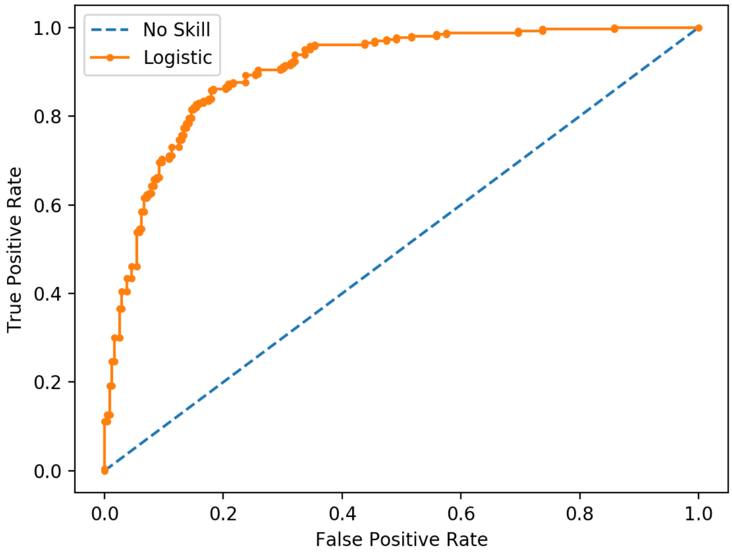


Figure 4.12: A ROC Curve plot for a generic logistic regression model (Brownlee, 2018)

area under the ROC curve.

Due to robustness and insensitivity to class imbalance, in addition to the measurement of classifier performance over different thresholds, AUC is chosen to be our primary method of comparing and ranking models in this thesis.

4.6 Model Interpretation

In the setting of non-parametric models, we are limited to comment on correlations of the features and not the causal relationships. As an example, whether the assignee of the patent is a Japanese individual or corporation may be correlated with patent litigation, but being a Japanese individual probably does not cause the patent in question to be litigated. Holding the features that affect litigation fixed, whether the assignee of the patent is a Japanese corporation or not, does probably not have any direct effect on the fact of litigation. The unobserved omitted variables may still be important for the result. Therefore, interpreting the coefficients causally should be done with caution.

Relative Variable Importance

In a data mining application, the features in the input data are seldom equally important (Hastie et al., 2009). In most applications, only a few of them have substantial influence on the outcome, while others can almost be excluded from the dataset (Liaw and Wiener, 2002). Relative importance is important to assess as it has an influence on the outcome and by understanding the relative importance, the model becomes more interpretable. The evaluation functions can be separated into two group: functions that utilized the model information, and functions that do not (Molnar, 2019). For classification trees relative importance of predictive variables are measured as the total decrease in node impurities from splitting on the variable, averaged over all trees (Liaw and Wiener, 2002). Gini index, presented in equation 4.11, is used as the measure for node impurity. Commonly, the output of variable importance is presented as bar graphs. The variables are ranked in order of decreasing importance.

Partial Dependence Plots

Partial Dependence Plot is a model-agnostic method to analyze and visualize the marginal effects of the variables have on the outcome of the model (Molnar, 2019). A PDP plot can show how the relationship between the target and a variable is, whether it is linear or more complicated. However, it is possible to identify the variables that seem to be related to the outcome of whether a patent will be litigated or not. An assumption for partial dependence is that the features are not correlated. Partial dependence works both for numerical and categorical features. Partial dependence marginalizes the output of the prediction model over the distribution of the features. The function of the partial dependence (4.20) represents the relationship between the features of interest and the outcome of the model. When marginalizing the other features, the function only depends on the chosen features.

$$f_S(X_S) = E_{X_C} f(X_S, X_C) \tag{4.20}$$

The X_S are the variables for which the partial dependence function is to be calculated and visualized. X_C are the other variable used in the machine learning model in focus. Due to the limitations of the computer graphics, usually, there are only one or two variables in the set S. The variable vectors X_S and X_C in combination make up the total feature space X. Partial dependence marginalizes the model output over the distribution of the variables in set C, so that thethe relationship between the features in set S we are interested in is shown. By marginalizing over the other features, we get a function that depends only on features in S, including the interactions with other features (Molnar, 2019).

The advantages of the partial dependence plots are that the computation is intuitive, the interpretation is clear and plots are easy to implement. However, there are several disadvantages. One disadvantage is that the realistic number of features that can be analyzed is restricted to two due to the 2- or 3-dimensional representation possibilities (Hastie et al., 2009). Further, the assumption of independence might not hold for all real datasets.

4.7 Tools

The models and data manipulation in this work is constructed in R programming language. Most models are constructed in the *caret* (Kuhn, 2008) library environment. To mitigate the computational limitations of ordinary desktop computers, virtual machines on Microsoft Azure and custom built computers are used to utilize on-demand capacity.

5 Analysis and Results

In this chapter, we carry out our analysis on the dataset presented in Chapter 3 using the methods presented in Chapter 4. First, we will explain the prediction framework and how the different models in the analysis are constructed. We then compare the predictive performance (mainly in terms of AUC) of the constructed models and other important metrics used in statistical learning.

5.1 Model Performance and Results

The models are able to predict patent litigation to a relatively large extent, where both the addition of more data and other algorithms yields an increase in predictive power. Models trained on a balanced dataset perform better, although we encourage future work to explore different ratios of sampling than the 50/50 ratio used in this thesis. In our analysis we comment on each model's performance and try to analyse why the models perform as they do.

For all models, the dataset is split in a 70-30 % allocation between training and test set. The data selection to both sets is made randomly across all cohorts ensuring as little bias as possible in the selection phase. The validation set used in training is generated with repeated cross-validation with 5 folds and 5 repeats which minimize both variance and bias in the training process without loosing a large amount of observations to evaluation. The training and tuning of the models are set to maximize model's performance measured in AUC.

5.1.1 Logit - The Baseline model

The baseline model, inspired by the model constructed by Lanjouw and Schankerman (2001), is a natural starting point in exploring the opportunities in prediction of patent litigation. The model, a logistic regression model, is chosen, not only because it is used by Lanjouw and Schankerman (2001), but also because it offers simplicity in terms of construction and intuition.

The base dataset, as discussed in Chapter 3 and shown in table 3.2, is used with the baseline model to produce a starting point. The initial baseline model yields following results:

Logit (1) base data				
Confusion Matrix			Metrics	
	Referenece		Accuracy	0.9918
Prediction	**Yes**	**No**	Sensitivity	0.0015
Yes	12	41	Specificity	0.9998
No	7,740	936,428	AUC	0.7635

Table 5.1: Model performance - logit I

The model achieves an accuracy of 0.9918, almost the same as the no-information rate for this dataset. The no-information rate is the accuracy rate that can be achieved without a model, by simply guessing the outcome to be of the major class (Kuhn and Johnson, 2013). The sensitivity is very low at 0.0015, with only 12 correctly classified positive cases and 41 negative cases predicted to be positive. This model is so conservative that it missed nearly all the positive cases and thus yield a high number false negatives. This is a common problem in classification of severely imbalanced datasets; the accuracy is high, but almost all cases of interest are missed. The model scores an AUC of 0.7635, which indicates some predictive performance in the model, but applied so conservatively that it is not useful in any practical business application, as it fails to correctly identify litigation cases.

To remedy the model's conservatism, we introduce sampling methods that balance the dataset before model training. In total three methods are considered; under-sampling; over sampling; and ROSE sampling. Over- and under-sampling is performed with the *caret* package (Kuhn, 2008), while the ROSE sampling is performed with the ROSE package (Lunardon et al., 2014). The results of the logit model with these sampling methods are shown in table 5.2

Results Sampling Methods - Logit - Base Data

Sampling method	Accuracy	Sensitivity	Specificity	AUC
Under-sampling	0.6836	0.7083	0.6834	0.7628
Over sampling	0.6849	0.7071	0.6844	0.7636
ROSE sampling	0.9918	0.0016	0.9918	0.7637

Table 5.2

The model trained on data sampled with ROSE performs much in the same way as the model did without sampling and offers no real improvement. Both under- and over-sampling perform differently. With these sampling methods, the model trades accuracy for increased sensitivity while AUC remains unchanged. As we are interested in the increased sensitivity, both methods offer an improvement. Of the two methods, we prefer the under-sampling as it performs equally as good as the over sampling and does not introduce extra, simulated or repeated observations into the training model. The drawback of under-sampling is that we fail to utilize the majority of observations obtained. However, the small difference in AUC suggests that we do not loose much predictive power in utilizing fewer observations.

Although, when using under-sampling, a number of real observations are removed from the training set, all the observations of the litigated patents are kept in the training set.

5.1.2 Logit with Extended Data

To study whether the performance of the logit model increases with more information, the baseline model is trained on a dataset with 13 new variables, in a model which we refer to as logit 2. The extended dataset is constructed of the indicators discussed in Chapter 3 and shown in table 3.2. Sourced variables with high correlation are removed prior to training. The model's parameters remain unchanged. The result of the logit 2 model with the extended dataset is shown in table 5.3

Logit 2 with extended data

Confusion Matrix			Metrics	
	Reference		Accuracy	0.7083
Prediction	**Yes**	**No**	Sensitivity	0.7234
Yes	5,608	273,289	Specificity	0.7082
No	2,144	663,179	AUC	0.7870

Table 5.3: Model performance - logit II

The second logit model scores higher both in terms of sensitivity and AUC, meaning that both the predictive performance increases and the rate of false negative predictions decreases. Thus, addition of more data has a beneficial effect for the model's performance and in the efforts to predict patent litigation. The number of false negatives at 2,144 is improved, but indicates that the model is still somewhat conservative in its allocation of the positive class.

5.2 Exploring other Algorithms

Although increased number of variables improved our classifier's performance, there is still room for improvement. We expand upon the work of Lanjouw and Schankerman (2001) by testing the performance and suitability of additional algorithms. We test three different algorithms that we believe to be suited to our prediction problem: Random Forest, XGboost, and Support Vector Machine.

5.2.1 Random Forest

In terms of complexity, decision trees are often regarded as simple approaches to classification (James et al., 2013). Moreover, decision trees have become increasingly popular when predicting on imbalanced data (Cieslak and Chawla, 2008). Therefore, in our opinion, tree-based models are a natural stepping stone in exploring alternative algorithms to the logit algorithm.

The first tree-based method is the Random Forest algorithm which is run with a parameter grid and under-sampling. The Random Forest model uses the same extended dataset as used in the logit (2) model.

Although Random Forest uses bootstrap data sampling, the model is trained with 5 repeated 5-fold cross-validation. This is probably redundant, and we suggest for future work that the necessity of the repeated cross-validation be evaluated in order to improve run-time. However, repeated 5-fold cross-validation is included in our study in order to give all models the same format of input.

Results Sampling Methods - Random Forest

Sampling method	Accuracy	Sensitivity	Specificity	AUC
Under-sampling	0.6973	0.8813	0.6958	0.8828
No sampling	0.9918	0.0000	1.0000	0.9245

Table 5.4

From table 5.4 we see many of the same patterns as we have seen in the logit models; the model trained on an under-sampled dataset are able to label litigation cases while reducing the number of false negatives, at the cost of accuracy. The model trained on the imbalanced data scores high in accuracy but fails to identify any positives. The accuracy of of the non-sampled model is 0.9918, which is the same as the no-information rate for our data-set. The predictive power of the Random Forest models are much higher than the preferred logit model. The AUC of the no-sampling model is 0.9245, which is much higher than any other model tested in this thesis, while the under-sampled model achieves an AUC of 0.8828.

The under-sampled model with a ROC-AUC of 0.8828 is, despite the higher AUC of the non-sampled model, our preferred Random Forest model, as the non-sample model fails to identify most litigation cases and achieves a sensitivity of 0.00.

Even though the non-sampled model is a no-information classifier, the results are encouraging for future work. There is probably more performance to be extracted from the Random Forest algorithm on this dataset if trained to respond better to class imbalance. We expect that introducing weights to the model or changing the ratio of sampling to a ratio other than 50/50 could be beneficial in improving the performance further.

Our Random Forest's hyperparameters are determined by a parameter grid in order to systematically search for the best combinations of hyperparameter settings. The optimal hyperparameter settings are shown in table 5.5.

The number of variables available for splitting at each tree node, controlled by the *mtry* parameter, is often recommended to be of size $mtry \approx \sqrt{p}$ where p is the number of variables in the dataset. In the dataset we have 21 variables giving a suggested *mtry* of approximately 4.58. The optimal *mtry* of 5 shows our model preforms the best with the recommended size.

Even though the split criterion is not a tunable hyperparameter as discussed in section 4.3.3, we include the randomized split criterion in the parameter grid as a categorical parameter. The Gini Index is selected to be the better option.

The optimal minimum node size is 1, which indicates a deep model inline with a typical random forest classifier. If the minimal size is higher, each individual tree can stop splitting at an earlier stage, making them shallow. Our model must conversely be regarded as a deep tree-model. Segal (2004) finds that increasing number of noise variables lead to a higher optimal node size, indicating that our model can handle any noise in our data.

Random Forest best model performance

Hyperparameters	
mtry	5
Split Criterion	gini
Min. Node Size	1

Confusion Matrix			Metrics	
	Reference		Accuracy	0.6973
Prediction	**Yes**	**No**	Sensitivity	0.8813
Yes	6,832	284,893	Specificity	0.6958
No	920	651,575	AUC	0.8828

Table 5.5

If we compare the confusion matrix of the best Random Forest model in table 5.5 to the confusion matrix of the second logit model in table 5.3, we see that the number of false negatives has decreased from 2,144 to 920 and true positives has increased from 5,608 to

6,832, which is favourable behavior. However, the Random Forest model as a decreased overall accuracy, which is the result of increased false positives. As we are seeking models that are mostly useful for insurers, this trade-off is beneficial.

The main improvement in the best random forest model is the large increase in predictive performance in terms of AUC. An AUC score of 0.8828 is an increase of 0.0959 from the second logit model. The extension of the dataset for the logit models yielded an increase in AUC of 0.0242. This suggests that there is much improvement in predictive ability to be made by employing an appropriate algorithm. This is also encouraging for further studies, as it is easier to change the means of statistical analysis than to change the amount and quality of data available.

5.2.2 XGboost

XGBoost is another decision tree method that has been heavily promoted in later years in the machine learning community. XGBoost offers a series of tunable hyperparameters that are chosen with a parameter grid. From table 5.6 we see that, similar to the Random Forest models, that the non-sampled XGBoost yield more predictive power, with an AUC of 0.8205, but in a no-information model. The under-sampled XGBoost is our preferred XGBoost model, even though it has a lower AUC score.

XGBoost Results

Model	Sampling method	Accuracy	Sensitivity	Specificity	AUC
XGBoost	Under-sampling	0.7082	0.7610	0.7078	0.8100
	No sampling	0.9918	0.0028	1.0000	0.8205

Table 5.6

The preferred XGBoost model is shown in table 5.9.

XGboost - hyperparameter settings	
Boosting iterations	600
Learning Rate	0.025
Tree Depth	6
Column Sampling	1
Row Sampling	1
Gamma	2

Confusion Matrix		
	Reference	
Prediction	**Yes**	**No**
Yes	5,899	273,669
No	1,853	662,799

Metrics	
Accuracy	0.7082
Sensitivity	0.7610
Specificity	0.7078
AUC	0.8100

Table 5.7: Model performance - XGBoost

In the confusion matrix of the preferred XGboost model in table **??** the number of false negatives is 1,868 and true positives has increased to 5,884, which is favourable behavior. However, the Random Forest model outperforms the preferred XGBoost model both in terms of the number of false negatives and in terms of predictive power. The AUC of 0.81 is an improvement on the logit 2 model, but fails to improve upon the Random Forest model.

As the XGBoost algorithm is flexible with many hyperparameter options, it is possible that we are unable to extract the best performance possible from our model. Although grid search was utilized, not all possible combinations for hyperparameters were tested. With long run-times, we find the XGBoost model to be at an disadvantage due to the increased number of hyperparameters to run in the parameter grid. We were unable to draw any benefit from the out-of-core feature in the XGBoost model, as our computers did not have sufficient RAM to be able to run these models.

5.2.3 Support Vector Machine

Support Vector Machine (SVM) is known to perform well in a variety of settings, and is considered by many as one of the best "out of the box" classifiers (James et al., 2013). We run four different Support Vector Machines in total; a SVM with a linear kernel with and without sampling; and a SVM with radial kernel with and without sampling. As discussed in 4.3.5, we choose to include both radial and linear SVMs as we are unsure whether the problem can be linearly separated.

Support Vector Machine Results

Model	Sampling method	Accuracy	Sensitivity	Specificity	AUC
SVM Linear	Under-sampling	0.7078	0.7334	0.7076	0.78923
	No sampling	N/A	N/A	N/A	N/A
SVM Radial	Under-sampling	0.7124	0.7371	0.7122	0.7873
	No sampling	N/A	N/A	N/A	N/A

Table 5.8

The main challenge of using SVMs on our dataset is that the implementation are computationally heavy, and require substantial run-times. Besides making SVMs impractical in business applications, the complexity of the model drives a need for system memory that exceed what is available in our work when run on the entire non-sampled dataset. Therefore, we are unable to study the differences in performance between sampled and non-sampled datasets in the SVM.

From table 5.8 we see that there is little difference in performance between the two models, suggesting that the problem is close to linearly separable. If we assess the preferred hyperparameter settings of the radial SVM in table 5.9, that the model's decision boundary, decided by the hyperparameter C, is small, and a large penalty is assigned to wrong classifications and margin errors. Gamma parameter defines how far the influence of training examples should reach. For the non-linear model, the optimal value of gamma is low, meaning the influence of the training examples reaches "far". In other words, the model's decision boundaries are not "very curvy", and might resemble the linear SVM not only in scored results, but also in its shape.

Although the performance is largely similar, we prefer the radial SVM as it achieves better accuracy, sensitivity and specificity. Moreover, as the radial SVM can behave much in the same way as the linear SVM and offers more flexibility, which might be beneficial in business application.

SVM with Radial Kernel				
Hyperparameters				
C		64		
gamma		0.0133		
Confusion Matrix			Metrics	
	Reference		Accuracy	0.7124
Prediction	**Yes**	**No**	Sensitivity	0.7371
Yes	5,714	269,543	Specificity	0.7122
No	2,038	666,925	AUC	0.7873

Table 5.9

The Radial SVM achieves better accuracy than all other under-sampled models tested in this thesis. However, it achieves practically the same AUC as the baseline Logit 2 model. The sensitivity is comparably mediocre, where it scores better than both logit models, but worse than the decision trees.

In part due to its complexity, another drawback of SVMs is the difficulty in interpretation, making them less suitable for business application. Based on the results of our SVM models, we do not recommend the use of SVMs in prediction patent litigation.

5.2.4 Comparative Performance - Model Assessment

Model	Sampling	Accuracy	Sensitivity	AUC
Logit (1)	under	0.6836	0.7083	0.7628
Logit (2)	under	0.7083	0.7234	0.7870
Random Forest	under	0.6973	**0.8813**	**0.8828**
SVM Radial	under	**0.7124**	0.7371	0.7873
XGboost	under	0.7082	0.7610	0.8100

Table 5.10: Performance of best model of each algorithm

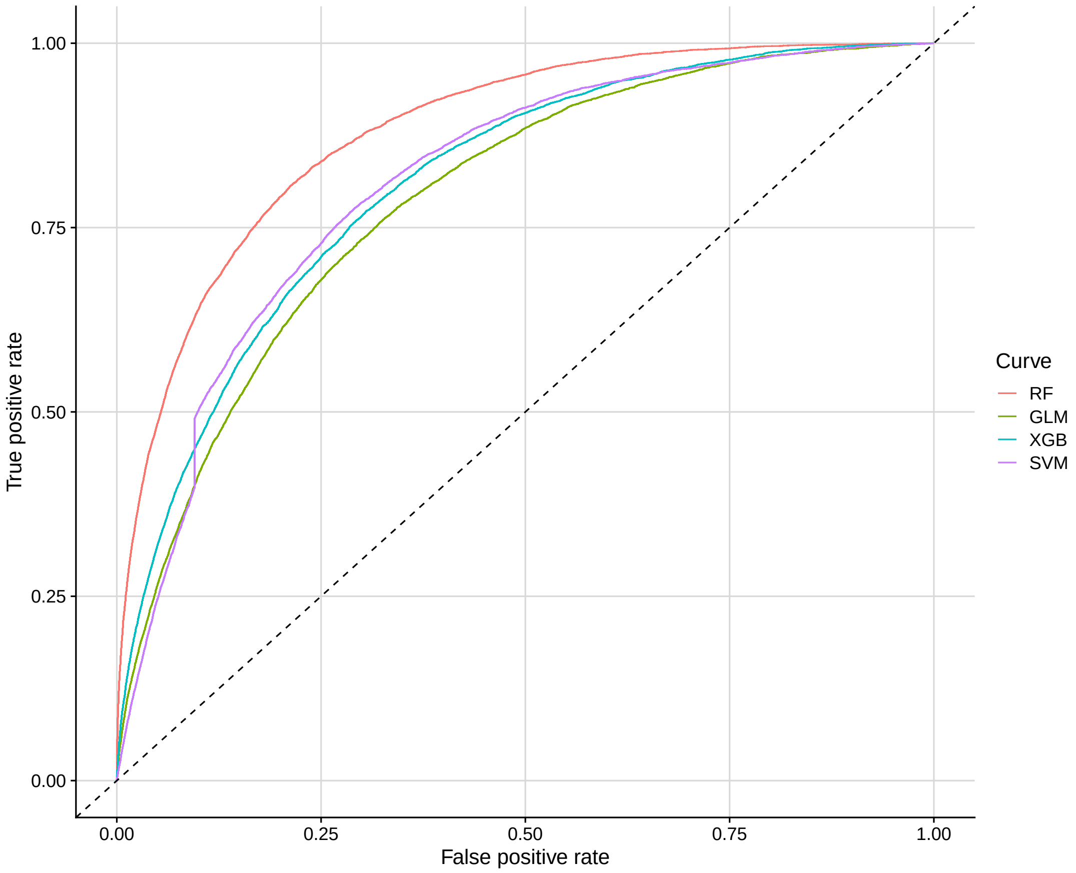


Figure 5.1: ROC curves for computed models

Table 5.10 shows the performance of the best models within each algorithm discussed in this thesis. The best performance of all algorithms are achieved from training the models on the balanced dataset, while training the models on the imbalanced dataset adds almost no more value to the application than the no-information rate. The Random Forest model with the under-sampled data outperforms all other models tested by a large margin. An AUC of 0.8828 makes the Random Forest model better in terms of predictive power with a margin of 0.0728 compared to the next best model XGBoost. In terms of sensitivity, which is used to measure the ability to minimize the number of false negatives is also much higher than any other model.

Figure 5.1 shows the ROC curves for the four different models, all trained on the under-sampled dataset. The curves indicates that the Random Forest model has larger predictive power and the ability to distinguish between true positives across most thresholds, and maximizing the difference in performance at approximately 0.25 false positive rate. It is also able to gain 100 % of positive prediction at an earlier threshold than the other models tested, indicated by the higher position of the Random Forest curve at the top-right corner.

Based on our findings, we prefer the Random Forest algorithm in predicting patent litigation on our dataset. It achieves more predictive power and fewer false negatives than any other model tested. A conceivable drawback that might affect its suitability in business application is it instability, where minor changes can affect performance. Furthermore, our findings indicate that the SVM algorithm is not suited for predicting patent litigation on our dataset, as it produces results that are as good as the baseline logit model while being much slower, more complex, and prone to exceeding the system memory capacity of our computers and virtual machines.

5.2.5 Alternative Algorithms

In our study we have tested several algorithms that we deemed to be of the same level of complexity and potential as the binary logistic regression model. Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) were both tested as viable options to logistic regression (logit). However, they both had similar performance as logit models, and did not offer any new insights. We therefore exclude these algorithms from our thesis. Another simple alternative to logit, the K-nearest neighbours (KNN) algorithm was considered. We were however not able to run the KNN-algorithm on the dataset due to computational limitations. Further, Generalized Additive Model (GAM) was tested as a non-parametric alternative to the logit model, but did not perform better either. Lastly, Neural Networks were tested as a promising, modern technique. However, the development of a Neural Network model required substantial parameter tuning beyond the scope of this thesis.

6 Discussion

As mentioned in Chapter 2, this thesis is partly inspired by the findings in "Characteristics of Patent Litigation: A Window on Competition" by Lanjouw and Schankerman (2001). Since Lanjouw and Schankerman wrote the paper in 2001, much has changed. The availability and potential quantity of data, as discussed in chapter 3.1, have increased massively. The advances in algorithms, statistical libraries, and methods have given us a wealth of possibilities and opportunities researching patent litigation. Moreover, the massive increase in computational power enables us to research much larger quantities of data. Although this affords us numerous opportunities and possibilities in researching and writing this thesis, it can also derail the thesis into an endless quest for complex algorithms and data expansion without discovering the insights being made. Thus, in exploring the opportunities of predicting patent litigation, we have committed to a fundamental heuristic of simplicity.

> *"Everything should be made as simple as possible, but not simpler"*
>
> *- Albert Einstein (Sessions, 1950)*

In considering the multitude of different opportunities in terms of both algorithms and data made available to us, we have sought to simplify in an Occam's Razor-esque approach to feature and method selection. “Entities are not to be multiplied beyond necessity" - we will not add complexity beyond reason or necessity.

6.1 Performance of the Prediction Models

As our baseline model, we replicate the model used by Lanjouw and Schankerman (2001) and study its performance. We then expand the dataset, adding more variables in order to study whether the predictive power of the model can be improved by introducing more information. A decision-tree model, in the form of the Random Forest Algorithm, is then introduced to assess the viability of tree-based models on our dataset and study the potential increase in predictive power. An alternative decision tree model, XGBoost, which uses boosting instead of bagging, is then considered.

In the interest of simplicity, we have sought to minimise the extent of model hyperparameter tuning. There are no single unified approach in parameter tuning, and by keeping the model relativily simple, it is easier to study what drives prediction performance and to assess the merits of each model. We recognize that flexible models such as XGBoost are constructed and run without reaching its fullest potential, but extensive hyperparameter tuning is outside the scope if this thesis.

6.2 The Value of More Information

In our study of predicting patent litigation, we constructed a baseline model inspired by Lanjouw and Schankerman (2001). The first order was to study the effect of increasing the number of indicators on model performance. The effect, in terms of change in predictive power, can be measured as the difference in performance between the two implementations of the logit model. From section 5.1.1 we see that the performance in terms of AUC increases with ≈ 0.024. This means that just by adding more variables to the dataset (the number of observations remains the same) we can increase the predictive power of our model. The added variables are all available at the time of grant and can be viewed as intrinsic characteristics. We expect that there is potential of increased predictive power if acquired characteristics are included. However, for our intended use-case we cannot include variables based on acquired characteristics as they are not known at the time of grant. Moreover, all of the accuracy-based metrics, and importantly sensitivity, improve. This means that some of the information added is useful in describing the factors causing litigation.

6.3 The Preferred Model

We find that the model that performs best overall with our dataset, for the purposes of this thesis, is the Random Forest algorithm trained on an under-sampled training data. This model achieves the highest predictive power while at the same time minimizing false negatives. The under-sampled Random Forest model achieves a ROC-AUC of 0.8828. The best model produces 920 false negatives, which is much better than the next best model with 1,868 false negatives. This is of course not perfect, as the goal is to construct a model that makes as few mistakes as possible while reducing false negatives. It is also interesting to note that the non-sampled random forest achieves an AUC of 0.9245 (!), albeit in a null-classifier model, which indicates that there is potential for better predictions within the model and the dataset, that can be unlocked.

The increase in predictive power achieved by using a different algorithm than the simple logit model used by Lanjouw and Schankerman (2001), is to us surprisingly large. The increase can be measured as the difference between the AUC of the logit 2 model and the best model, which is approximately 0.0958 AUC. What makes this particularly interesting is the fact that the SVM and XGBoost, powerful algorithms in their own right, fails to achieve even half the performance increase on the logit 2 model. This indicates that employing a more powerful algorithm in itself does not realize the full potential of the predictive power in a dataset, and that it is more important to employ the *right* algorithm with appropriate tuning.

One of the advantages of decision tree methods is that they are very intuitive in function and are easy to understand and to explain (James et al., 2013). This might be beneficial in light of our use-case, where insurers might deploy the method when evaluating an insurance application. The ease in understanding the model, its method, and the output might be preferable for the potential end-user.

Another advantage that might make tree-based models particularly suited in our case is that they can handle qualitative variables (of which we have many) without the need to create dummy variables (James et al., 2013). Recall that the logit model needs to split the tech field variable into 33 binary dummy variables. A decision tree model does not need to do so, which is beneficial for model performance, and also provide the opportunity to test more elaborate categorical variables, such as higher granularity in IPC tech field classes.

It is not immediately clear to us why the Random Forest model performs better than the XGBoost model. Presented in section 4.3.4, the XGBoost algorithm should perform well in cases with characteristics in our dataset, but is outperformed by the "simpler" Random Forest algorithm. They are both tree models that utilize CARTs to make their predictions. Both models are run with 1,000 trees, which should negate some of the difference in how they are trained in terms of weak vs strong learners. A key strength of the Random Forest algorithm is that it restricts the number of available variables as split candidates to a randomly selected subset. This prevents the dominance of strong predictors in top level splits and decorrelates the trees. It is possible that due to the relative difference in importance between variables included in the models, the Random Forest is able to utilize more information, by including different variables in the top levels. In fact, in figure 6.1 we see that the relative importance is dominated by a few variables.

6.4 Variable Importance - Characteristics and Risk

In identifying and discussing the importance of different characteristics and risk factors, we use relative variable importance. Relative variable importance ranks variables according to the decrease of Gini impurity when a variable is chosen as split candidate. Another tool is partial dependence, that measures to marginal effect of a variable on the predicted outcome. As these measures do not infer importance in itself and they are not based on coefficients, we cannot determine statistical significance. These metrics merely give a picture of what characteristics and variables are important in the model's predictions. As our preferred model is non-parametric, these metric remain as useful tools in determining risk factors and importance of characteristics. The results are useful in discussing patent litigation risk factors, but should be interpreted carefully.

In the plot in figure 6.1 the Y-axis ranks the variables in order of decreasing importance. The values on the X-axis the relative importance of the variables. The variables are important factors for the outcome of the models. However, it is important to keep in mind that this importance must be reviewed in combination with the other variables used in the dataset. In the plots in figures 6.2 and 6.3 the values on the Y-axis indicate the relative logarithmic contribution of the values of the variable on class probability. The negative values indicate that the negative class is more likely for the observation, and vice versa. By assessing the partial dependence plots we cannot state definitively that one variable leads to increased or decreased risk of litigation, and as they are one-dimensional they do not capture co-dependence in importance.

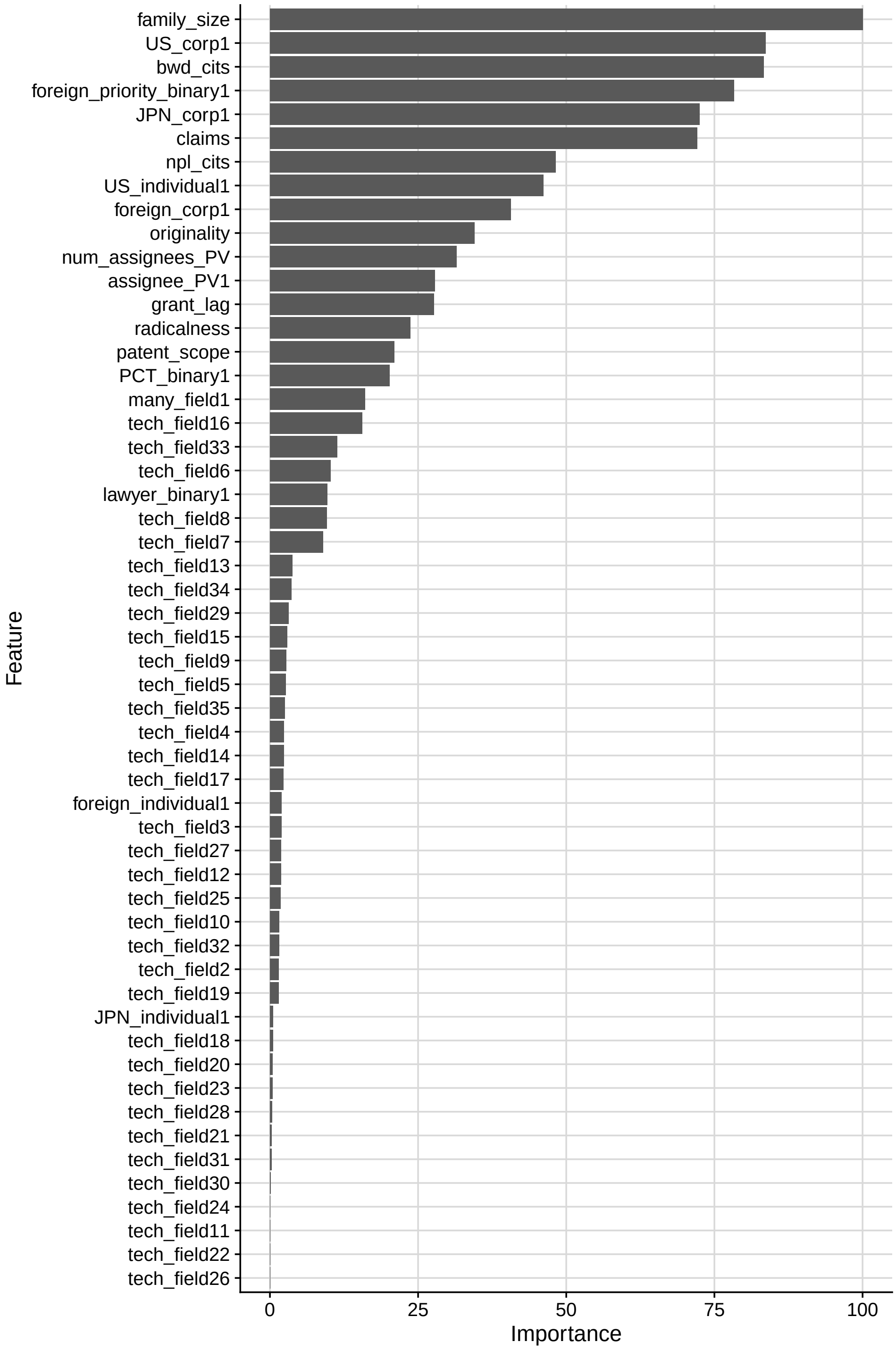


Figure 6.1: Relative Importance of variables

Family Size

The most important variable in the random forest model is the family size variable (figure 6.1). The family size variable denotes the number of jurisdictions and patent offices a patent is filed in. If we assess the partial dependence plot in figure 6.2, we see that a larger family size is associated with an increase in the probability of an observation being litigated. The $\hat{y}$ (yhat), which denotes the predicted label of an observation by the model, is positive when there is an increased probability of an observation being positive, and negative when there is a decreased probability. Patents having less than approximately 5-6 family size have a negative $\hat{y}$ while for any number of that there is a positive and increasing $\hat{y}$. This supports the findings of Chien (2011) and Cremers (2004) which links the family size with patent value and risk of litigation.

US Corporation

The second most important variable is the US corporation variable. Being a US corporation is associated with an increased risk of litigation. The top-right graph in figure 6.2 shows that US corporations have a positive probability of being litigated, while not being a US corporation is associated with a greater decrease in probability. This seems to contradict the findings of Lanjouw and Schankerman (2001) that individuals are more prone to litigation than corporations. However, being a US individual is also associated with a positive $\hat{y}$, although considered to be less important in terms of relative importance. The importance of US-corporation, foreign priority, and US individuals might suggest that domestic (US) patents that are more likely to be litigated.

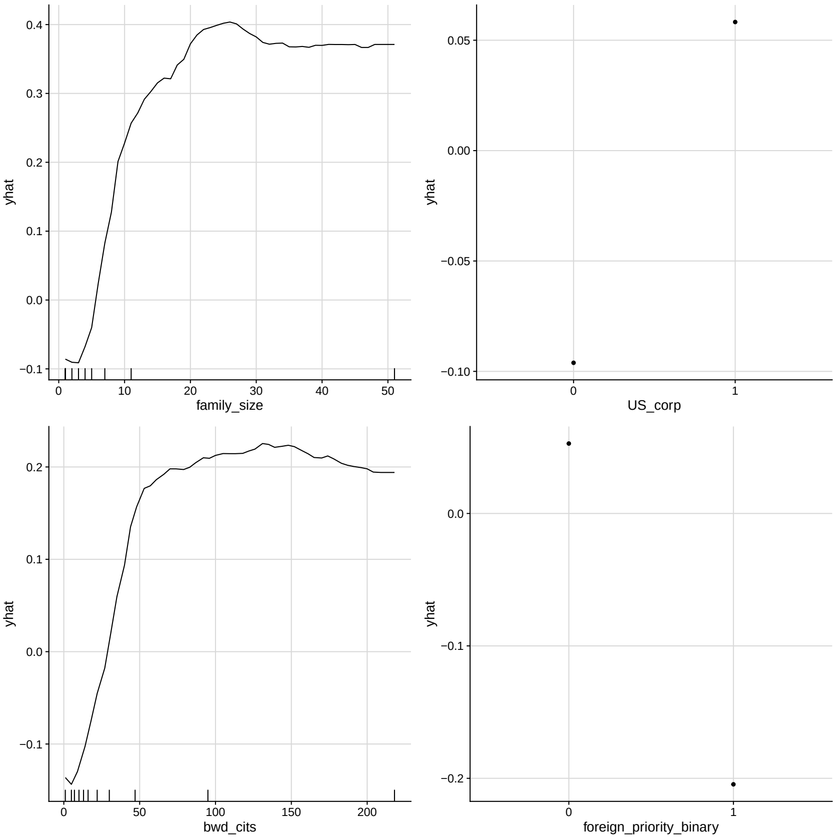


Figure 6.2: Partial Dependence Plot for the 4 most important variables of the Random Forest Model

Backward Citations

The third most important variable is the number of backward citations. A larger number of citations to previously granted patents, is associated with increased risk of Litigation. Chien (2011) finds also that the number of backward citations are important indicators of litigation risk. By assessing the bottom-left graph in figure 6.2, we can see that the numbers between 0-30 are negatively associated with $\hat{y}$, meaning reduced probability of litigation. However, in numbers larger than 30 there are positive values of $\hat{y}$ in an increasing trend.

Interestingly, Lanjouw and Schankerman (2001) find, although not significant, that the increase in backward citations is associated with a decrease in litigation risk. Lanjouw and Schankerman state that less citations indicate a new field of technology and thereby increased risk of litigation, which can be confirmed by their findings of an inverse relationship. Our results suggest the opposite. Our findings lends itself to Squicciarini et al. (2013) which finds that the number of backward citations is an indication of quality, where quality is measured as value. More valuable patents tend to be more litigated as they are worth defending or acquiring (Chien, 2011). Our findings on backwards citations support a direct relationship between the number of backward citations and litigation risk.

Foreign Priority

The foreign priority variable is another important variable for the preferred Random Forest model. Patents that have not been previously filed in another country, and thus not been given priority by the USPTO, are associated with decreased probability of being litigated. If we recall from table 3.3 60.06 % of all patents do not have foreign priority. It might be several reasons as to why the foreign priority variable is so important for the best model. Patents first filed abroad then filed in the US might indicate that the patent owner is both affluent and ambitious for its patent and accordingly might display some of the deterrence characteristics described by Lanjouw and Schankerman (2004). In any case, patents granted foreign priority have a lower probability of being litigated. Reviewed literature has, to our knowledge, not used foreign priority as a characteristic in patent litigation prediction.

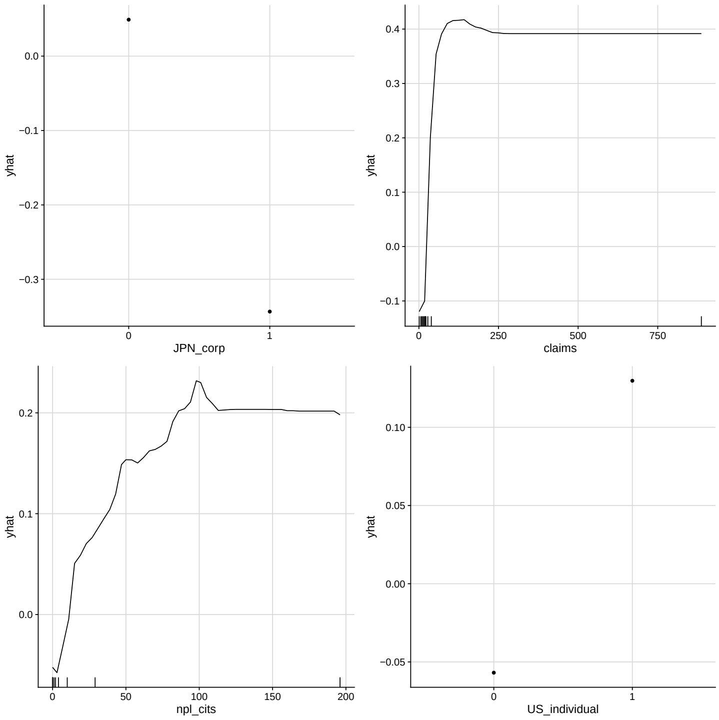


Figure 6.3: Partial dependence plot for important variables in the Random Forest Model

Japanese Corporation

The Japanese Corporation binary variable is also important for the model. If the assignee of the patent is a Japanese Corporation the $\hat{y}$ is strongly negative, meaning that the probability of the patent being litigated is much lower than patents with other types of assignees. This lends itself to the idea that patents assigned to foreign entities have decreased risk of litigation.

Number of Claims

The number of claims is another important variable for the model. The number of claims is generally considered an indicator of patent complexity. However, the patentee has many incentives to claim as much as possible and it is up to the patent examiner to question different claims. One might argue that broad and complex patents are more likely to be litigated as they encompass more areas, ideas, and industries, thus increasing the exposure to the risk of conflict with other entities and patent holders. These findings might be of interest to patent offices and examiners. Patents that cover more areas might also be viewed as more valuable, and thus more exposed to litigation. It is important to note that the distribution of claim (depicted by the black stripes along the X-axis in figure 6.3) is left-skewed and the curve of partial dependence needs to be interpreted carefully. Both Lanjouw and Schankerman (2001) and Chien (2011) finds that claims are important indicators.

Non-patent Literature Citations

Higher numbers of non-patent literature (NPL) citations indicate increased probability of being litigated. For numbers between 0-25 $\hat{y}$ is negative, while for higher numbers $\hat{y}$ is positive. This is in line with findings by Branstetter (2005) that NPL citations have significantly higher quality than patents that do not cite scientific literature.

US Individual

Interestingly, US individual scores high on relative variable importance. The variable increase the probability of litigation when true (1). If we compare the ratios of $\hat{y}$ increase between US individual and US corporation in figure 6.3 and 6.2 respectively, although the

US individual variable is less important in creating splits in the model, it has a stronger effect on $\hat{y}$. This supports our hypothesis that it is not necessarily the difference between being an individual or a corporation that is important for predicting litigation, but rather whether or not the patent assignee is domestic or foreign.

Technology Fields

The variable "Technology fields" is perhaps the most recognisable differentiating characteristic of a patent, as it describes what is being patented. Intuitively, some technology areas such as pharmaceuticals or semiconductors, should carry more risk of litigation than others. However, assessing the relative variable importance graph in figure 6.1, none of the technology fields are deemed as particularly important for the model. The most important technology field is Pharmaceuticals which scores approximately 15 % in relative importance. This is far less than most of the other variables and characteristics used. The relative unimportance of technology fields might be due to the low granularity used. The IPC technology field system has many more detailed levels that might be more suitable for describing litigated patents. It seems that in our study that variance within a technology field is greater than the variance between technology fields. Therefore, in our study technology field is not a useful indicator for predicting patent litigation.

The patent scope variable, which uses the number of IPC technology fields recorded in the patent, is similarly not an important variable for our model, even though Squicciarini et al. (2013) and Lerner (1994) associate patent scope with patent value. Another technology field based variable that in unimportant for the model is the Radicalness index, further signalling the inappropriateness of IPC-technology fields as predictor of patent litigation. The PDP plot for the technology fields is presented in Appendix A0.2.

Other Variables

Grant lag, the time between application and grant, is not very important for the model. This is perhaps somewhat surprising, as grant lag is associated with patent value as documented by Squicciarini et al. (2013); Harhoff and Wagner (2009); and Régibeau and Rockett (2010).

Whether the patent is owned by an individual or an assignee, measured in the binary assignee variable, is moderately important for the model, scoring around 25 in relative variable importance. Likewise, the number of assignees are somewhat important with a score of approximately 30. Partial dependence suggests that the number of assignees is inversely related to the probability of litigation risk, where more assignees reduces the risk. Interestingly, having an assignee is associated with a reduced probability of litigation which does not support the findings of Lanjouw and Schankerman (2001), but is more aligned the findings of Cremers (2004).

6.5 Applications

We find that our preferred model, the Random Forest model, can predict patent litigation to a large extent. We believe that the inclusion of the variables discussed in this thesis combined with what we find to be the most appropriate model, yields predictions that can be useful in business applications.

The AUC measure is useful in measuring the increase in information, as any score above 0.5 is better than randomly guessing and a score of 1 indicates perfect information. With our preferred model, the Random Forest, we are able to achieve an AUC of 0.8828. Although not perfect, the model introduces more information into patent litigation, which could be made available to decision makers. In studying to what extent patent litigation can be predicted, the AUC score is the clearest measure, as it is connected to the predictive power of the model.

6.5.1 Use-case and Incentives

The primary use-case in our study is that of mitigating the risk of insuring patents for litigation. We believe that our preferred model is suitable to be used in this setting, as it provides high sensitivity in the form of few false negatives, and high predictive power.

In the insurance setting, the two parties involved, insured and the insurer has different incentives in weighting prediction error type. By utilizing predictive models in the insurer context, the insurance companies can adjust their litigation insurance premiums based on percieved risk. The insured party will benefit from prediction models that minimize false positives since a lower number of predicted positive cases most likely reduce insurance premium costs (at least in the short term). The insurer on the other hand has opposite incentives, as they seek to minimize false negative predictions. The insurer would in most cases prefer to sell insurance to patent owners that are predicted to be litigated (with an insurance premium that reflects this) rather than sell insurance to a patent that is predicted to not be litigated but is litigated (false negative). Thus, the end-user of the model has different incentives in selecting how model should be tuned and selected.

The preferred Random Forest model is suitable for the interest of the insurer: it has few false negatives at the cost of many false positives. An insurer might use our model to gauge the risk associated with insuring a particular patent, as there is a relatively good chance that if the model outputs no litigation for a given patent. The model successfully identifies 88 % of all litigation cases.

Other applications

We hope that our findings also can be useful in other use-cases, such as patent portfolio management and for patent offices. Our findings indicate that for the patent data we have collected in conjunction with the prevalence rate of litigation, that the use of the Random Forest algorithm can generate predictions of value and can adapt to changes. This performance is independent of what type of error is being measured, and we expect that the model performs just as well on false positive minimization and will retain the relative high AUC scores.

Through relative variable importance for the model, we are more able to understand what drives the prediction of patent litigation. This can be useful in business application as the characteristics are often easy to recognize and understand. For quick reviews, an insurer, portfolio manager, or patent office can assess the family size, if the patent is domestic or not, and the number of backwards cites. Our insights on variable importance are not a substitute for predictions, but can help users understand what to look for in more risky patents. Our findings on tech-field importance can also be of value, as patents belonging to technology fields perceived to be less risky, can be equally at risk as patents belonging to other fields.

Predictions powered by our preferred model can be used by patent portfolio managers to monitor the litigation risk of any patent in their portfolio. However, for continuous monitoring we expect that the inclusion of acquired characteristics, as discussed by Chien

(2011), would be beneficial. Nonetheless, our findings indicate that the Random Forest algorithm is suitable in the patent portfolio management use-case, and are optimistic of even greater predictive power if acquired characteristics are utilized.

For national patent offices, we hope our best model can be useful in evaluating litigation risk. Since a court case for patent infringement or other disputes also is a cost for society, the national patent offices partly have a responsibility to prevent such costs. By utilizing predictive models in the process of patent application assessment, the patent office can get a prediction on the possibility for the patent in focus to be litigated. With predictive models, the examiners at the patent offices possibly can prevent that by assessing the characteristics of the patent, such as the number of claims or citations.

6.6 Robustness and Critique

6.6.1 Limitations and Challenges

A challenge with prediction on patent data is that the properties of the patents can be changed over time. Thus, some patents that appear to have a high probability of litigation will avoid litigation through some form of external assistance. This can act as a disruptive element in our models and reduce the ability to predict. Again, as mentioned in section 3.6, the under-reporting of the data is a challenge for constructing models that reflect the real world perfectly.

Our models perform poorly at extreme values of the explanatory variables. However, this is natural and predictive power is not significantly affected as this applies to very few patents.

Real-life implementation will meet some distinct challenges. Legislation, processes, and incentives can change over time and the effects are challenging to implement into a statistical model. However, predictive power is achievable, and with some adjustments, methodical hypotheses can be developed. This will produce a value for reducing risk and predicting possible litigation, but will not give an ultimate answer to the life cycle of a patent.

Computational Challenges

The amount of data used for predicting is large and handling of that amount of data requires extensive processing power. Especially computing SVMs has been challenging. Support Vector Machines are models that are suitable for data classification and are valuable because of the good generalization performance. Although, SVMs are able to avoid the "curse of dimensionality" effectively, said ability leads to an increase of sample size or dimensions which cannot be solved (Wang et al., 2016). Therefore the computational time is high and requires much computational power that might not be widely available.

Feature Selection

Feature selection is a process in which the less relevant variables are excluded from the datasets that the models are trained and tested on. There are several methods in doing so, and in order to avoid evaluating methods of feature selection in this thesis, feature selection has not been performed. Therefore, the evaluation of the models and their performance heavily relies on the assumption that the features chosen for the dataset consists of characteristics that have an effect on a patent's probability to be litigated. Highly correlated variables are dropped initially and represent the extent of feature selection in this thesis.

Limitations of Parameter Tuning

The methods tested in this thesis are flexible methods, with possibilities to tune the hyperparameters to achieve adjusted models for different applications. Although parameter grids were used to find the optimal combination of parameters, the truly optimal combinations are probably not found. The chosen hyperparameters must be regarded as an estimate. In addition, some of the chosen parameters have little impact on the performance of the models and the values of those are therefore arbitrary.

6.7 Further Research and Future Work

In this thesis we have focused on evaluating the performance of different models on our patent litigation dataset. The algorithms chosen are by no means all that are available.

Other gradient boosting algorithms such as *catBoost* or *LightGBM*, or many of the options within neural networks can be explored.

As stated earlier, our main heuristic was that of simplicity. Our work should be viewed as an exploratory study in the suitability of different models in their basic form. As such, all models tested could probably be improved through tweaking and tuning. This is especially the case with neural networks, a powerful family of algorithms which were tested but not used in our study, which can be tuned extensively. We have chosen not to explore the potential of these algorithms in depth, as demands a lot of resources without any guarantee of improvement.

From our findings there are mainly two subjects that we would first recommend to be studied further and where we see the most potential in gains: weighting and resampling for Random Forest; and stochastic XGBoost. As presented in section 5.2.1, we produce a Random Forest null-classifier with a AUC of 0.9245. The high AUC signals that there is potential for an increase in predictive power in the preferred model. Weighting in the model or different ratios of training/validation set balancing might unlock the potential in a model that is useful. We are optimistic that there is potential for improvement on our preferred model.

As the realtive variable importance plot (figure 6.1) shows that a few variables are considered more important than the majority, we believe that a stochastic XGBoost variant could improve the performance of the XGBoost model as it mimics the random split candidate feature of the Random Forest algorithm.

As mentioned in section 4.3.2, decision trees can utilize categorical variables without the need of creating dummy variables. Therefore they can be run with a much greater number of different categories within a categorical variable. Finding that the Random Forest algorithm is the most appropriate model enables the use of high-level categorical variables. In our research we have been limited in the number of IPC-technology classes available, as the non-decision tree models were not able to run with more levels. The effect on performance by adding IPC-technology on a higher granularity could be studied.

Another technique and focus of the analysis could be Survival Analysis, often referred to as Duration Analysis in economics. In survival analysis the patent litigation is evaluated on of how long it takes before a patent is litigated and what describes those events. Even though the assumption of the fact that all patents will be litigated at some point might not hold, it is an interesting approach to analyze some other aspects of the lifetime of a patent.

We recommend for future work to evaluate feature selection, as some variables might be too noisy and thus reduce the performance of the models. Furthermore, the reduction of the number of the features can improve the running time and shrink the computational complexity of the data gathering, data processing and model construction.

There is a wealth of available data to be explored and tested. We expect that patent litigation prediction can be improved by the use of characteristics freely available but not discussed in this thesis.

7 Conclusion

In this thesis we have sought to explore the extent to which patent litigation can be predicted such that it can be used to improve the functionality of the intellectual property ecosystem. Inspired by Lanjouw and Schankerman (2001), who studied characteristic importance and risk factors of litigation through logistic regression, we first sought to emulate their studies in terms of indicators and characteristics chosen and method employed. However, in order to make predictions of value to decision makers at time of grant (in particular insurers), some characteristics used by Lanjouw and Schankerman are excluded as they are simply not known at the time.

The general goals of this study have been to understand the effect of introducing more characteristics, more data, and to study the benefits of utilizing more appropriate classifiers than the traditional logit classifiers. Moreover, we have sought to understand which characteristics are important for model decision and thus by analogy risk factors for patent litigation.

In order to measure the performance and tune models based on usefulness for stakeholders and entities that have actual need of litigation prediction, all models where measured based on their ability to correctly identify litigation cases (through AUC) and at the same time minimize the number of false negatives (through sensitivity).

Our work is related to anomaly detection, and as such, we found that in order to avoid null-classifiers, all models performed the better when trained on an under-sampled training set. We find the method of under-sampling to be the preferred sampling option, as the dataset is large enough to allow the reduction in observation available in training. Furthermore, we expect that increase performance can be extracted by exploring other sampling rations than the 50/50 ratio used in this thesis.

We find that patent litigation can be predicted to a large extent when appropriate methods and algorithms are employed. The predictive power of a logit model, as used by Lanjouw and Schankerman (2001), can be improved by introducing variables available in the OECD patent quality dataset (OECD, 2020) and the PatentsView data base (USPTO, 2020). The gain in predictive power by including more variables is approximately 0.02 AUC. We expect that this increase can be improved by adding more variables that are important for litigation prediction.

In order to explore the possibility of improving predictions through an increase in predictive power, three additional classifiers where tested. We find the Random Forest algorithm to produce the best results on our dataset both in terms of predictive power and the number of false negatives produced. The Random Forest model achieves an AUC of 0.883, an increase of 0.125 from the baseline logit model. XGBoost performed better than the base-line logit model, but did not outperform the Random Forest model. We discourage the use of SVM on highly imbalanced patent litigation data as it is computationally heavy and fails to produce satisfactory results.

Furthermore, we find that although the Random Forest algorithm yields the highest prediction scores on our data, the results can be unstable. The instability of the Random Forest model needs to be accounted for in its use. We find that the other algorithms produce less accurate but more stable results.

The most important characteristic that drives the prediction of litigation is the family size variable. The family size variable describes the number of patent offices a patent is registered at. Where a larger family size is associated with increased risk. The binary variable US corporation is the second most important variable, where not being a US corporation is associated with a reduced risk of litigation. The third most important variable is the number of backward citations, where an increase in the number of citations is associated with an increase in litigation risk. In general we found that there is some ambiguity in whether or not an individual or corporation is associated with increased risk. However, it seems that domestic (US) patents are more exposed to litigation than their foreign counterparts, and that this trait is more important than whether or not the patent is owned by a company. Moreover, we find the characteristics and variables based on technology fields are not important for the prediction of patent litigation.

We find that patent litigation can to a large extent be predicted and that the Random Forest algorithm is particularly well suited for prediction. Challenges relating to class imbalance makes the prediction of patent litigation difficult, but can be overcome by appropriate methods and sampling. We are optimistic that our results can be improved upon be more extensive work and hope that our research can contribute to better predictions of patent litigation and remedy some of the uncertainty in the intellectual property ecosystem and preserve the incentives for innovation.

References

Akosa, J. (2017). Predictive accuracy: a misleading performance measure for highly imbalanced data. In *Proceedings of the SAS Global Forum*, pages 2–5.

Ali, A., Shamsuddin, S. M., and Ralescu, A. L. (2015). Classification with class imbalance problem: A review. *International Journal of Advances in Soft Computing and its Application*, 7(3).

Bender, G. A. (2000). Uncertainty and unpredictability in patent litigation: The time is ripe for a consistent claim construction methodology. *J. Intell. Prop. L.*, 8:175.

Bloebaum, S. (2007). Past the tipping point: Reforming the role of willfulness in the federal circuit's doctrine of enhanced damages for patent infringement. *NCJL & Tech.*, 9:139.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145 – 1159.

Branstetter, L. (2005). Exploring the link between academic science and industrial innovation. *Annales d'Economie et de Statistique*, pages 119–142.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 26(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brownlee, J. (2018). How to use roc curves and precision-recall curves for classification in python. Retrieved from ://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/?fbclid=IwAR0N74YGSjYZp92IfPvr67Am2927t-A8Wy4npAVPn8eeCiUyn1RcocRAmPc.

Chabacano (2020). Own work, cc by-sa 4.0.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y., and contributors, X. (2020). Introduction to boosted trees. Retrieved from https://xgboost.readthedocs.io/en/latest/tutorials/model.html. (Accessed on 03/09/2020).

Chien, C. V. (2010). From arms race to marketplace: the complex patent ecosystem and its implications for the patent system. *Hastings Lj*, 62:297.

Chien, C. V. (2011). Predicting patent litigation. *Texas Law Review*, 90(2):283–329. Copyright - Copyright University of Texas, Austin, School of Law Publications, Inc.

2011; Document feature - Charts; Tables; Graphs; ; Last updated - 2012-01-16; CODEN - TXLRA2; SubjectsTermNotLitGenreText - United States–US.

Cieslak, D. A. and Chawla, N. V. (2008). Learning decision trees for unbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 241–256. Springer.

Cremers, K. (2004). Determinants of patent litigation in germany. *ZEW-Centre for European Economic Research Discussion Paper*, (04-072).

Cremers, K., Harhoff, D., Scherer, F., and Vopel, K. (2003). Citations, family size, opposition and the value of patent rights. *Research Policy*, 32:1343–1363.

Efron, B. and Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pages 54–75.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874. ROC Analysis in Pattern Recognition.

Feick, L. (2019). Evaluating model performance by building cross-validation from scratch. Retrieved form https://www.statworx.com/de/blog/evaluating-model-performance-by-building-cross-validation-from-scratch/. (Accessed on 04/02/2020).

Flach, P. A. and Wu, S. (2005). Repairing concavities in roc curves. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, page 702–707, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Ganglmair, B., Helmers, C., and Love, B. J. (2018). The effect of patent litigation insurance: Theory and evidence from npes. *Available at SSRN 3279130*.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

Guts, Y. (2018). Target leakage in machine learning. Retrieved from https://image.slidesharecdn.com/targetleakageinmachinelearning-181029144130/95/target-leakage-in-machine-learning-4-638.jpg?cb=1540824167. (Accessed on 06/15/2020).

Hagiu, A. and Yoffie, D. B. (2013). The new patent intermediaries: platforms, defensive aggregators, and super-aggregators. *Journal of Economic Perspectives*, 27(1):45–66.

Harhoff, D. and Wagner, S. (2009). The duration of patent examination at the european patent office. *Management Science*, 55(12):1969–1984.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer New York.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*, volume 112. Springer.

Kapil, A. R. (2018). K-fold cross-validation. Retrieved from https://www.datavedas.com/k-fold-cross-validation/. (Accessed on 05/14/2020).

Kesan, J. P., Schwartz, D. L., and Sichelman, T. (2011). Paving the path to accurately predicting legal outcomes: A comment on professor chien's predicting patent litigation. *Tex. L. Rev. See Also*, 90:97.

Khalilia, M., Chakraborty, S., and Popescu, M. (2011). Predicting disease risk from highly imbalanced data using random forest. *BMC medical informatics and decision making*, 11:51.

Kim, J.-H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational statistics & data analysis*, 53(11):3735–3745.

Klabjan, D., Wongchaisuwat, P., and McGinnis, J. O. (2017). Predicting litigation likelihood and time to litigation for patents. page 257–260.

Kleinbaum, D. and Klein, M. (2010). *Logistic Regression: A Self-Learning Text*. Statistics for Biology and Health Series. Springer.

Koyejo, O., Natarajan, N., Ravikumar, P., and Dhillon, I. (2014a). Consistent binary classification with generalized performance metrics. *Advances in Neural Information Processing Systems*, 3:2744–2752.

Koyejo, O. O., Natarajan, N., Ravikumar, P. K., and Dhillon, I. S. (2014b). Consistent binary classification with generalized performance metrics. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2744–2752. Curran Associates, Inc.

Kuhn, M. (2008). Building predictive models in r using the caret package. *Journal of Statistical Software, Articles*, 28(5):1–26.

Kuhn, M. and Johnson, K. (2013). *Applied Predicitve Modeling*. Springer.

Lanjouw, J. O. and Schankerman, M. (2001). Characteristics of patent litigation: A window on competition. *The RAND Journal of Economics*, 32(1):129–151.

Lanjouw, J. O. and Schankerman, M. (2004). Protecting intellectual property rights: are small firms handicapped? *The journal of law and economics*, 47(1):45–74.

Lerner, J. (1994). The importance of patent scope: An empirical analysis. *RAND Journal of Economics*, 25(2):319–333.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23.

Lunardon, N., Menardi, G., and Torelli, N. (2014). ROSE: a Package for Binary Imbalanced Learning. *R Journal*, 6(1):82–92.

Maier, R. (2020). Recent takes from the supreme court and federal circuit on attorney fees awards in patent cases. Retrieved from https://www.law.com/newyorklawjournal/2020/01/21/recent-takes-from-the-supreme-court-and-federal-circuit-on-attorney-fees-awards-in-patent-cases/?slreturn=20200423092456. Accesed on 04/23/2020.

Marco, A. C. and Tesfayesus, A. (2017). Patent litigation data from US district court electronic records (1963-2015). *SSRN Electronic Journal.*

Menardi, G. and Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28(1):92–122.

Molnar, C. (2019). *Interpretable Machine Learning.* https://christophm.github.io/interpretable-ml-book/.

Petherbridge, L. (2011). On predicting patent litigation. *Tex. L. Rev. See Also*, 90:75.

Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1301.

Provost, F. and Fawcett, T. (1999). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 43-48.

PwC (2018). 2018 Patent Litigation Study. Retrieved from https://www.pwc.com/us/en/forensic-services/publications/assets/2018-pwc-patent-litigation-study.pdf. (Accessed on 05/12/2020).

Régibeau, P. and Rockett, K. (2010). Innovation cycles and learning at the patent office: does the early patent get the delay? *The Journal of Industrial Economics*, 58(2):222–246.

Ripley, B. D. (2007). *Pattern recognition and neural networks.* Cambridge university press.

Schölkopf, B., Burges, C., and Vapnik, V. (1996). Incorporating invariances in support vector learning machines. In *International Conference on Artificial Neural Networks*, pages 47–52. Springer.

Segal, M. R. (2004). Machine learning benchmarks and random forest regression.

Sessions, R. (1950). Retrieved from https://www.nytimes.com/1950/01/08/archives/how-a-difficult-composer-gets-that-way-harpsichordist.html.

Spackman, K. A. (1989). Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 160 – 163. Morgan Kaufmann, San Francisco (CA).

Squicciarini, M., Dernis, H., and Criscuolo, C. (2013). Measuring patent quality.

OECD (2020). Oecd patent quality indicators database january 2020.

US Department of Justice (2020). 220. attorney's fees. Retrieved from https://www.justice.gov/jm/civil-resource-manual-220-attorneys-fees. (Accessed on 04/23/2020).

USPTO (2020). Patentsview. Retrieved from https://www.patentsview.org/web/. (Accessed on 02/19/2020).

WIPO (2020a). About the international patent classification. Retrieved from https://www.wipo.int/classifications/ipc/en/preface.html. (Accessed on 04/25/2020).

WIPO (2020b). Protecting your inventions abroad: Frequently asked questions about the patent cooperation treaty (pct). Retrieved from https://www.wipo.int/pct/en/faqs/faqs.html. (Accessed on 04/25/2020).

Trajtenberg, M., Henderson, R., and Jaffe, A. (1997). University versus corporate patents: A window on the basicness of invention. *Economics of Innovation and new technology*, 5(1):19–50.

USPTO (2020a). General information concerning patents. Retrieved from https://www.uspto.gov/patents-getting-started/general-information-concerning-patents. (Accessed on 04/25/2020).

USPTO (2020b). Patent cooperation treaty. Retrieved from https://www.uspto.gov/patents-getting-started/international-protection/patent-cooperation-treaty. (Accessed on 04/25/2020).

USPTO (2020c). Patent litigation docket reports data. Retrieved from https://www.uspto.gov/learning-and-resources/electronic-data-products/patent-litigation-docket-reports-data. (Accessed on 01/20/2020).

USPTO (2020d). Right of priority of foreign application. Retrieved from https://www.uspto.gov/web/offices/pac/mpep/s213.html. (Accessed on 04/25/2020.

Wang, X., Huang, F., and Cheng, Y. (2016). Computational performance optimization of support vector machine based on support vectors. *Neurocomputing*, 211:66 – 71. SI: Recent Advances in SVM.

Appendix

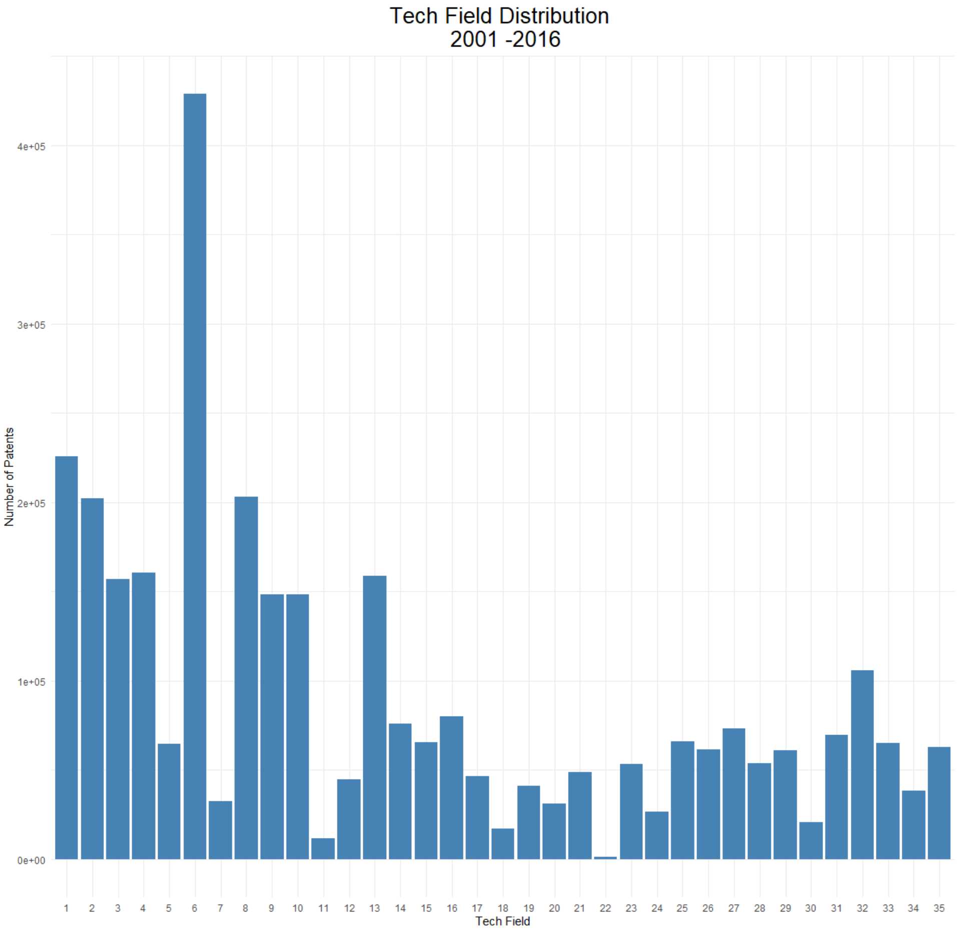


Figure A0.1: Distribution of tech field over all patents 2001-2016

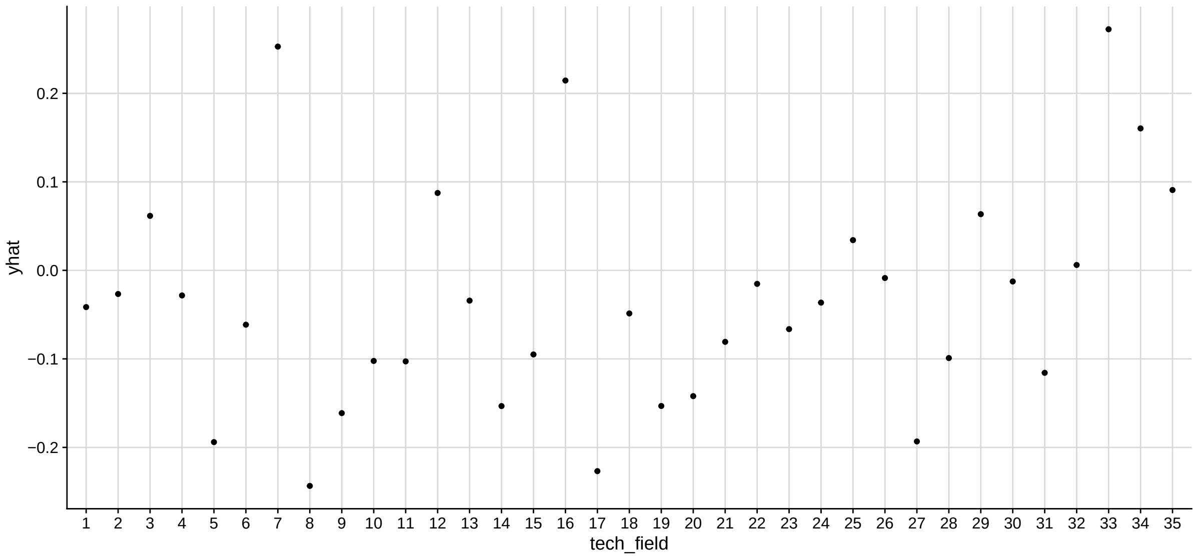


Figure A0.2: Partial Dependence Plot for technology fields

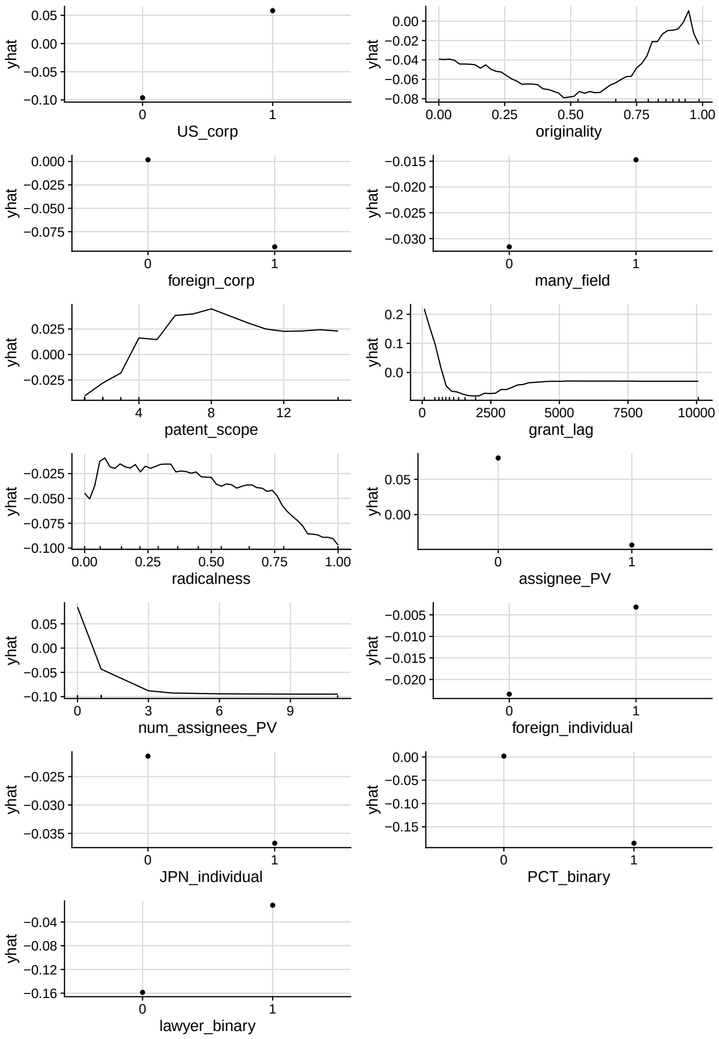


Figure A0.3: Partial Dependence Plot for remaining variables