Jheronimus Academy of Data Science

Master's Thesis

Prediction of Adolescent Criminal Law Case Outcomes Using Unstructured and Structured Data

Author:
Nazlı Ander

Primary Supervisors:
Prof. dr. F.J. Bex
dr. D. Kolkman
Practical Supervisor:
dr. N. Tollenaar

A thesis submitted in fulfillment of the requirements
for the degree of
Master of Science in Data Science and Entrepreneurship

May 22, 2019

Abstract

Machine learning methods have been adopted by a variety of real-world commercial applications including optimization of Internet search results, facial recognition and fraud detection. However, legal professionals and researchers lag behind in their adoption of these tools. Considering the time-consuming nature of quantitative analysis in the legal domain, machine learning methods can provide a systematic and efficient way to model the judicial processes. Following the previous studies on predictive quantitative analysis in the legal domain, we constructed a machine learning setting to explore to what extent we can automatically predict the outcomes of Adolescent Dutch Criminal Law cases (whether they were decided as Juvenile Law or Adult Law). To predict the outcomes we used unstructured (judgment text) and structured data (database features) relevant for 499 criminal law cases. While combining unstructured with structured data we experimented dimension reduction techniques for text. We used machine learning pipelines to run the experiments. We defined three main paths: Text only (Path 1), Database features only (Path 2) and Combinations of Text and Database features (Path 3). For Path 1 and Path 3 we introduced a dimension reduction option for text features. A series of experiments on a balanced dataset (under-sampled) were run using three different classification algorithms, consisting of one boosting (Light Gradient Boosting) and two linear algorithms (Logistic Regression and Support Vector Machines). Although by using all the available information we were able to predict with 100% average Accuracy, similar to the previous predictive studies that use before trial information to model judicial decisions, we reported 76.5% as our best average Accuracy and 81.79% as best average AUC-ROC. Our findings indicate that only the boosting algorithm benefits from dimension reduction techniques for text when we combine unstructured and structured data. Future studies are recommended to improve the quality of the datasets and to experiment more on classification on an imbalanced dataset.

Keywords: machine-learning, dimension reduction, text mining, judicial decisions, criminal law

Acknowledgements

Many people and organizations contributed this thesis (project). I am very grateful to the people who directly or indirectly contributed to my research.

I would first like to thank my thesis supervisors Prof. dr. Floris Bex and dr. Daan Kolkman for guiding me throughout this project with their invaluable expertise. I appreciate the time and effort you invested. I am glad to have the chance to work with both of you.

I would like to acknowledge dr. André van der Laan and dr. Nikolaj Tollenaar for offering to work on an interesting task and giving birth to the compelling research questions answered in this thesis. I want to thank for your excellent feedback and helping me to interpret the results for criminal law.

The results that are described in this thesis are based on the dataset that WODC provided to me. Besides that, I am very grateful for the time that the WODC researchers spent with me to help me out during my research. As a result of our fruitful discussions, I was able to learn so much about ethics and criminal law.

While designing the experiments described in this thesis I received many helpful and creative comments from the NLP Thesis Cycle Group in Jheronimus Academy of Data Science. I explicitly thank Nitin Narayan for providing me Dutch judiciary word vector representations (Word2Vec) that are a part of his research project.

Before starting this research, I was contributing to a recommendation systems project in a web platform, Werkspot. Software development experience that I had there helped me to overcome all the technical difficulties with easiness while writing this thesis. I would like to thank Winfred Peereboom, Sacha de Rosner, Vijay Kiran, Alejandro Barba, Liveris Avgerinidis and all the other colleagues who showed me a way to program better data pipelines.

I received so much support for proofreading from my friends and family members. I would like to thank Aslıhan Akdeniz, Hande Sodacı, Ezgi Mamus, Vera Westra and George Avadan for their critical feedback and support to improve this technical text.

Lastly, I would like to thank to my parents and friends for their endless support and encouragement, in particular to my high school dormitory friends who have always been a part of my family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Age has been considered as a plausible predictor of crime independent of the social and cultural conditions, across countries and identical through time (Greenberg, 1977; Hirschi & Gottfredson, 1983). Given the official data, the plot of the official crime rates against age shows a sharp increase in the late adolescent years and decreases by the early 20s (Moffitt, 1993). The resulting plot has a robust shape, which is shown in Figure 1.1 for the Netherlands. Since the plot shape does not change as a function of countries and time, age is the most significant factor that can be considered for researching the increase of the crime rates in the adolescent years (Ulmer & Steffensmeier, 2014).

Age in relation to crime has been studied by a variety of disciplines, including sociology, psychology and biology. Similarly, there are diverse perspectives in the criminology literature that investigate the relationship between the age and criminal behavior. The social perspective focuses on the potential rewards from the social and economical environment, e.g. identity-claims, autonomy, economic security and peer-influences, as the driving factor of the criminal behavior in young adulthood (Steffensmeier, Allan, Harer, & Streifel, 1989; Steinberg, Cauffman, & Monahan, 2015). Psychological perspectives, on the other hand, explain criminal behavior in adolescence with the abilities such as the emotional maturity, decision making and risk taking (Hagan, 1991; Farrington, Loeber, & Howell, 2012). Unlike the social science perspectives, biological approaches explain the criminal behavior in adolescence with biological measures such as brain development, change in physical strength and impulse control(Farrington et al., 2012). Although those three perspectives provide different reasoning for the same outcome (i.e., the increase in criminal activities in the late adolescent years), they all agree on that these factors do not identically evolve for each individual.

The peak of the criminal rates in adolescence does not only concern researchers but also policy makers. By focusing on the well-adjusted policies about the prosecution processes of young adults, the recurrent criminal activity (i.e. recidivism) can be reduced (Council of State Governments Justice Center, 2015). Especially in

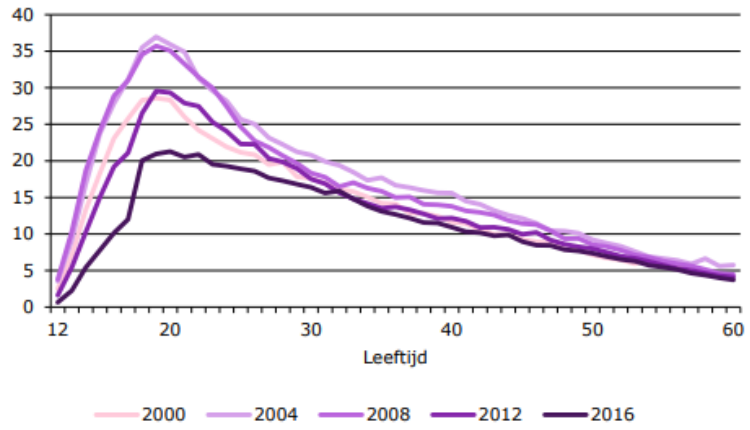


FIGURE 1.1: The Age and Crime Rate plot for the Netherlands, between the years 2000 and 2016. Source: Adapted from van der Laan and Beerthuizen, 2018.

the adolescence, offenders with different personalities respond differently to the correctional treatments (Glaser, 1974; Sherman, 1993). By following an individualized and tailored approach, the young adults' personal development and education can be targeted to fight with recidivism.

The Netherlands was one of the first countries to have a criminal procedure that is tailored to juveniles (Weijers & Liefaard, 2007). In 1905, the penal law Child Act entered into the Dutch law. It was based on the assumption that the court's decision should be guided by the child's future. However, the personality and the responsibility of the young offender had not been considered yet at that time. The upper age limit to which the penal law Child Act could be applied was 18, as from a legal perspective a child is considered as adult in his/her eighteenth birthday. The upper age limit was used as the main factor to decide between the individuals who could or could not be accountable for their criminal behavior.

In 1965, a revision in the juvenile justice system was made, by taking the juvenile delinquency into account. With the revision, the Child Care and Protection Board was established with an advisory and supervisory role for the juvenile law cases. Moreover, two exceptions on the upper age limit of the applicability of the juvenile law were introduced. Those were (i) the transfer of juveniles to the Adult Criminal Law (art. 77b, Dutch Criminal Code) and (ii) the transfer of young adults to Juvenile Criminal Law (art. 77c, Dutch Criminal Code). The exception (i) allowed, the juvenile court to sentence a juvenile who was aged 16 or 17, at the time of the committing offense, under the Adult Criminal Law. The exceptions were imposed based on the request of the public prosecutor, the severity of the offense and the personal circumstances of the offender. On the other hand, the exception (ii) allowed, the criminal courts to sentence young adults, who were aged 18, 19 or 20 at the time of the crime, under the Juvenile Criminal Law. The application of the exception (ii)

was less known and had been rarely used until the 2000s.

Throughout the history of Dutch Criminal Law, there were three Committees advised for the young adult law, Overwater-Committee (1951), Wiarda-Committee (1971) and the Anneveldt-Committee (1982). Both Wiarda and Anneveldt committees addressed the specific position of young adults or old juveniles (Liefaard, 2016). The Wiarda-Committee, proposed to lower the age limit from 21 to 18, because they assumed that 18 is the age where young adults gained independence from their parents (Wiarda-Committee, 1971). Differently, the Anneveldt-Committee stressed the need of a separate justice system for the young adults (Anneveldt-Committee, 1982). The committee suggested that the line between juvenile and the adult justice system must be drawn at the age, where juveniles are no longer predominantly influenced by the youth subculture. They stated that 24 was the age, where the young adults' tendency of delinquent behavior diminishes. Hence, they proposed to have a separate justice system for the young adults between the ages of 18 and 24. However, the Ministry of Justice rejected this offer with the concern of feasibility.

On April 1, 2014, with a new revision on the juvenile justice system, the upper age limit for adolescence was increased from 21 to 23. This revision was named as *Adolescent Criminal Law*, concerning the young adults that were aged between 18 and 22 at the time of the crime. With an emphasis on the educational development of the young adults, the Adolescent Criminal Law (art. 77c, Dutch Criminal Code) suggests that depending on the judicial conditions (maturity of the offender or the circumstances in which the offense is committed), it is possible to sentence a young adult (between the ages 18 and 23) under either Juvenile Criminal Law or Adult Criminal Law. The possibility of being sentenced under the Juvenile Criminal Law, for an adult that is older than the age of 18, could be considered as an exception to the standard rule of sanctioning young adults. The exception enables less matured young adults to benefit from a more pedagogical approach since juvenile justice system focuses more on education, rehabilitation and parenting (Barendregt & Laan, 2018). The Figure 1.2 shows the current situation of exceptions (i) and (ii).

The Scientific Research and Documentation Center (WODC) is an independent part of the Dutch Ministry of Justice. Within a 5-year research program, started in August 2014, the WODC monitors and evaluates the application of the Adolescent Criminal Law. To have a better monitoring and evaluation of the application of the recently changed Adolescent Law exception (ii), the WODC would like to have a machine learning model correctly predicting the decisions of the Adolescent Law cases. Their goal is to use the model deployed in a database for their monitoring research. Their current registration system is not complete and reliable. Therefore, they want to have a model with an accuracy that is closer to 100%. Besides, they would like to examine the judicial sentences with an automated method to understand the decision process of the judges. While providing an automated analysis on the decision process, it is improper to use all the available information including the

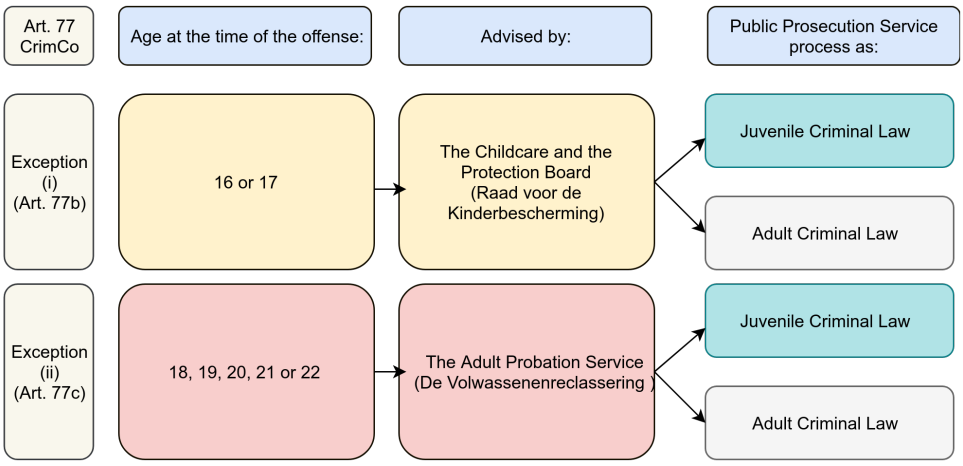


FIGURE 1.2: After the revision in 2014, the exceptions in the Art.77, Dutch Criminal Code.

decisions of the criminal law cases.

Based on the given facts, in this thesis we would like to use all the available information related to the Adolescent Law cases to have an accuracy score closer to 100%. At the same time we would like to have more insights on the decision process of the judges by comparing different kinds of information sources. Those two main goals contrast with each other, as one of them aims to use all the available information and the other one must be restricted with before decision information. Hence, we would like to define the questions to be answered in this thesis in two main sections: Research Questions and Practical Questions. Research Questions will be addressed to demonstrate the out-of-sample accuracy of the Adolescence Criminal Law cases, whereas the Practical Questions will determine the technical goal that is aiming to increase the accuracy. Those two will be discussed in detail in the following subsections. Then we will provide a simple outline of the thesis.

1.1 Research Questions

Although behavioralist law suggests that there is a personal, sociological and psychological element in judging, in theory judicial decision making must rely on a structure, that provides a degree of regularity (Gillman, 2001). The structure is most commonly represented as a casual reasoning between evidences and outcomes.

Social scientists have been interested in explaining the causal reasoning of the judicial decisions for decades. Causal reasoning between evidences and the outcomes can be described as mathematical computations, inductive rules or an explanation (story) (Pennington & Hastie, 1992). All the three descriptions of casual reasoning try to generalize the judicial procedure of evidence evaluation and finding the best fitting verdict as an outcome.

As the decision making process of judgment is expected to be structured, it can

be modeled manually by using legal experts or automatically via machine learning methods (Moses & Chan, 2014). By using a machine learning method, one can understand to what extent we can mimic the judicial decision process by using the information that decision makers could considerably have had before the trial started. In this thesis we will call the prior knowledge on the trials as *before trial information*.

Much of the available literature on machine learning in the legal domain focus on predicting the case outcomes by using the before trial information to provide assistance for the law professionals. Aletras, Tsarapatsanis, Preoţiuc-Pietro, and Lampos (2016) use before trial sections of the European Court of Human Rights (ECHR) judgment texts (unstructured) to predict the outcomes as violated or not violated. While conducting the experiments they automatically extract the topics in text to remove the redundant information occurring in the judgments. This process is called as dimension reduction in the machine learning literature. Likewise Katz, Bommarito II, and Blackman (2017) use a machine learning model to predict the outcomes of the United States (US) Supreme Court cases by using the database information (structured). We will provide detailed information about the predictive analysis in the legal domain in the Section 2.1.2.

Based on these examples, in this thesis we aim to understand the predictive power of the before trial information relevant for the adolescent criminal law cases. Different from the prior examples we will combine the unstructured and structured data while using common dimension reduction techniques for text as well. Therefore, the main research question of this thesis is as follows:

- To what extent, we can automatically predict the adolescent criminal case outcomes by using different combinations of **the before trial** sections of the judicial texts and different types of the database information?

Each criminal case involving an adolescent must be considered individually by the judges to determine whether the Juvenile or Adult Criminal Law to be applied, because, young adults do not grow at the same pace, and their psychological maturity levels are not identical. The judges give their decisions based on their previous experience with the similar cases, the pedagogical needs of the offender and the risk of recidivism. Since the decision process requires expert advice from the public institutions (Public Prosecution Service) and a law enforcement inquiry, it is not easy to come up with a predictive system to determine whether a given adolescent case sentence resulted in the Juvenile or the Adult Criminal Law just before the trial.

Knowing that both text information and the database information have different characteristics, we will treat those separately first to have a reference point. Then we will combine those two to validate whether the combination of both provides more information for the classification algorithms. As we try to answer the main research question, four sub-questions will be addressed to understand the underlying mechanism of correctly predicting the outcome of the case text and database information:

- Does the combination of different types of before trial sections in the judicial text improve the classification of the young adult cases?
- Does the combination of different categories of before trial database information improve the classification of the young adult cases?
- Does the combination of judicial text and structured information improve the classification?
- Do dimension reduction techniques for text improve the classification?

The questions will be answered by following a series of experiments with different classification algorithms, that are commonly used in the classification problems involving text. The outcomes of the Research experiment results are expected to be contributing to the Predictive Analysis in the Legal Domain academic literature.

1.2 Practical Question

Machine learning algorithms enable various text-intensive tasks to be automated. Machine learning methods have been adopted by a variety of real-world commercial applications including optimization of Internet search results, facial recognition, fraud detection, traffic predictions, stock market trading and medical diagnosis. Although, the law domain requires more advanced models than currently available, it lags behind in comparison to many other industries, when it comes to adopting the ingenious methods in computation and information systems (Surden, 2014).

To conduct scientific research, the WODC has access to many data sources relevant to security and justice, which are available in a structured database. On the other hand, many of the judgment text is publicly available in the Dutch Judiciary System.

Currently, by using an expert system in the structured database, it is possible to know whether the Juvenile Law has been applied to the young adults or not. The expert system consists of a logical structure, and it combines the application of each punishment type with *AND* or *OR* to come up with a classifier that distinguishes Juvenile Criminal Law from the application Adult Criminal Law. However, the expert system that the WODC is currently using performs better than a random guess (Accuracy of 70.5% and F1 score of 58.2%). The expert system that the WODC is currently using is discussed in detail in the Section 3.3.3. Because it uses manually entered punishment type data retrieved from the Dutch Public Prosecution Service (Openbare Ministerie), and those manual entries create noise in the dataset. Some of the machine learning methods are helpful for reducing the noise in the dataset, which do not necessarily improve the out-of-sample test accuracy. Those are generalized in the feature selection and dimension reduction algorithms. Given the above, by

using all the available information regarding the Adolescent Criminal Law cases and machine learning methods, it is possible to improve the predictions.

The main technical goal of this thesis is to develop an automatic analysis for the judicial cases regarding the Adolescent Criminal Law (art. 77c, Dutch Criminal Code) using features constructed on unstructured and structured data that is available for the relevant cases. In this thesis the potential of using combined text information and the structured data for the classification of the adolescent criminal law cases will be explored. By doing that we want to analyze to what extend the judicial cases are predictable, by using both before and after trial information available for the criminal law cases. Based on this, the practical (business) question that we aim to answer in this thesis is as follows:

- To what extent, we can automatically predict the adolescent criminal case outcomes by using judicial texts and the available database information?

The question will be answered by developing a machine learning application using all the available information, including the punishments and the decisions of the criminal cases. While combining the structured and unstructured data for predicting the outcomes of the adolescent criminal law cases. The machine learning application will be tested by running a series of experiments. If the results are satisfying for the WODC, we aim provide the model for the automation of the database labeling.

1.3 Outline

The types of the structured information will be defined as the Personal Information of the offender, the Procedural Information, the Criminal Information and the Punishment Information of the related cases. Additionally, four different sections of the judicial text will be used: the Indictment, the Evidence Considerations, the Punishment Motivations and the Decision sections of the criminal case judgments. The judicial text will be transformed into word and word pairs. The most popular dimension reduction techniques will be used to reduce the noise emerging with the high dimensional of the text data. All those information categories will be treated separately when we examine the sub-questions stated above. Then, while answering the last question, we will combine the text and the database information to examine the added value of using different sourced information.

To answer the Research Questions, we will not use all the available information for criminal case observations. In the experiments (aiming to answer Research Questions) we will use the information, which judges know **before** they start making their decisions in the trial process. Therefore we do not expect to predict all the observations correctly. However, this will add more academic value to this study instead of constructing a machine learning model while using all the available information.

The outline of this thesis will be as follows. A detailed literature review about the machine learning application in the legal domain and combining unstructured with structured data will be provided in the Chapter 2. Following that we will introduce the dataset that we used in this thesis in the Chapter 3. By using the dataset specifications we will define the methods to be used in detail in the Chapter 4. We will represent and comment on the Research Results in the Chapter 5. In the end we will provide the Practical Results and Conclusions in the Chapter 6 and Chapter 7 respectively.

Chapter 2

Related Work

To the extent of our knowledge there is no prior work that combines textual and structured information in the legal domain for applying machine learning methods. There are legal domain articles that use machine learning for quantitative analysis. Besides, there is research on the predictive effects of the combination of the textual and structured information. Therefore, the literature review is divided into two distinct categories: automated analysis in the legal domain and combining unstructured with structured data for machine learning applications.

2.1 Automated Analysis in the Legal Domain

To traditionally analyze judicial contents, it is required to *code* the judicial text for pre-identified content elements. Based on a research question, the coding is done by manually assigned by humans who have already analyzed the text according to a theory (Evans, Mcintosh, Lin, & Cates, 2007). To code, law students should be trained. Although the coding of legal documents is an important part of the systematic content analysis, the resulting coded pieces are reported not to be reproducible and to be prone to manual mistakes (Hall & Wright, 2008). However, computational approaches are fast becoming a key instrument in the legal domain as the number of articles that are highlighting the importance of automated analysis in legal studies is growing. The automated analysis in the legal domain can be separated into to subsections as non-predictive analysis and predictive analysis. We do not distinguish between the text and structured information as an input for the automated analysis, as long as they perform machine learning techniques.

2.1.1 Non-predictive Analysis in the Legal Domain

Non-predictive analysis in the legal domain covers the automatic information extraction from legal texts and similarity analysis. Automatic information extraction aims to automatically produce a shorter text from a long legal document. This is possible by extracting the topic representations of text and assigning a score to each

sentence in a text document to indicate its importance (Aggarwal & Zhai, 2012b). Furthermore, the similarity analysis aims to provide relevant prior cases for a given legal case automatically. This is possible by using distance and angular based measurements and network based analysis (Mandal et al., 2017).

In the automatic information extraction literature, Farzindar and Lapalme (2004) use the judgments of the Federal Court of Canada to apply text summarization based on the thematic segmentations in the legal documents. They use different parts of the judgments to assign argumentative roles while summarizing the legal documents. They present the resulting summaries in a tabular format divided by the thematic segments in the judicial documents. Likewise, Hachey and Grover (2006) use machine learning methods to create a work-flow that predicts the rhetorical status of legal case text and to select the most summary-worthy sentences from a document. In their work Chieze, Farzindar, and Lapalme (2010) create a framework for an automatic summarization and information extraction for legal information (ASLI). In this framework, they use symbolic methods to match the legal case sentences in the pre-defined sections for creating an automatized summarization. Following the work of Chieze et al., Yousfi-Monod, Farzindar, and Lapalme (2010) use machine learning to extract the sentences from the judgments, and match those with the possible summary sections of a judgment. Their results reported to be significantly improving the ASLI method.

In the legal case similarity subject, Olsen and Küçüksu (2017) provide a network analysis for the European Court of Human Rights (ECHR) and its Article 14 case law. They use citations to analyze how cases are clustered together by sharing citations. As Article 14 is about discrimination of the minorities, their work investigate the existing patterns of discrimination cases in the ECHR. In another study, Mandal et al. (2017) apply neural network based models to measure the similarity between the court case documents of the Indian Supreme Court. They experiment the Latent Dirichlet Allocation (LDA) to extract topics and compare the similarity performance with their experimentation with embedding-based methodologies such as Word2Vec and Doc2Vec.

2.1.2 Predictive Analysis in the Legal Domain

Much of the available literature on machine learning applications in the legal domain deals with the question of predicting the outcomes of cases. In their pioneering work, Martin, Quinn, Ruger, and Kim (2004) see the prediction of future events as the best test for an explanatory theory. They estimate multiple decision trees to predict the outcomes of the US Supreme Court decisions and compare the estimated model predictions with the legal expert predictions. Although their work does not contain a detailed statistical learning methodology, the model that they use performs better than the legal experts when predicting the US Supreme Court outcomes. Their

finding confirms Dawes (1979)'s findings that suggest statistical models are better than human intuition.

In their recent study, Katz et al. (2017) use an extensive set of categorical variables and Random Forest as a classifier to predict the voting behavior (whether they affirmed or rejected) of the US Supreme Court and its Justices. Their extensive set of categorical variables consist of the available features from the US Supreme Court (raw binary variables), features engineered that are derived from chronologically-oriented features and features engineered to summarize the behavior of a Justice, the Court, the lower court and differences between them. They achieve 70.2% level of accuracy on the case outcomes level and 71.9% on the justice votes level.

In another research, Șulea, Zampieri, Vela, and van Genabith (2017) aim to predict the law area of a case and the court ruling by using the legal text from the French Supreme Court. As they investigate the text features on the predictions of the French Supreme Court cases, they benefit from the Natural Language Processing methods as Bag of Words (BOW) as well. To reduce the dimensions and construct a directly linkable feature set, they use correlations with the labels as a feature selection method. Their model achieves more than 90% of accuracy for the law area predictions.

In their study, Aletras et al. (2016) use Natural Language Processing and Machine Learning to automatically analyze the ECHR cases, that belong to the Articles 3, 6, and 8. They use the specific sections of each ECHR case to correctly classify the outcomes (whether they were violated or not). While classifying the outcomes of the selected cases, first they extract topics for each case by using a topic modeling algorithm to reduce the number of the dimensions and prevent over-fitting. Their topic modeling algorithm is based on constructing a similarity matrix based on frequency matrix of text data and applying spectral clustering. Their results show that they obtain the highest accuracy scores when they use topic modeling and the circumstances section of judgements for all the Articles 3, 6 and 8. They report 79% as their highest average accuracy score.

Following the work of Aletras et al. (2016), Medvedeva, Vols, and Wieling (2018) use again different sections of the ECHR case laws to predict the outcomes (whether violated or not). Unlike Aletras et al., they exclude the Law section while validating their machine learning approach. Moreover, they increase the amount of the Articles of the ECHR and the cases per Article to be tested. They additionally analyze the weights assigned to each phrases by using the coefficients of the classification model that they use. Their models are able to predict decisions correctly with 75% accuracy on average.

As the most relevant literature for our thesis is predictive machine learning for law domain, while answering the research questions we will be referring to the research methods and outcomes stated in this section.

2.2 Unstructured with Structured Data for Machine Learning

Combining different sources of data creates many problems such as naming and structural conflicts. Naming conflicts arise, when different sources use different names for the identical attributes. This is easy to handle by changing the column names. However, structural conflicts occur in many different variations, such as different component structure, different data types and different integrity constraints. Those conflicts are harder to detect and solve (Rahm & Do, 2000). In the real world applications, with the increasing amount of web and Internet of Things (IOT) software, many data intensive applications using different data structures need to communicate with each other. To do that, they need to solve the structural conflicts. Much of the current literature on real-world data challenges pays particular attention to the class imbalance problems and unequal misclassification costs. However, there is still only a small amount of research that focuses on the practical issues in the real-world machine learning applications (Weiss, 2009).

There are two main approaches to combine different data sources for machine learning applications. Firstly, one can combine different classifiers for one or more of the data sources to classify new examples. Previous studies have reported a significant increase in performance, when different classifiers are combined to solve a particular classification problem (Xu, Krzyzak, & Suen, 1992; Dietterich, 2000). Secondly, one can use a series of transformations. Those cover pre-processing, dimension reduction, feature selection and clustering techniques (Han, Pei, & Kamber, 2011). The pre-processing of the different types of sources include aggregation, concept hierarchy generation and feature construction methods.

To the extent of our knowledge, there is not much prior work that combines text and structured information to the legal domain. However, this is not something new for the other domains, e.g., financial economics, web search engines and recommender systems. The most common research of combination of the text and structured information, in other words heterogeneous data, is done in the field of web recommender systems.

In her thesis van 't Hul (2018) uses the text classification results (probabilities) of the research of Kos, Schraagen, Brinkhuis, and Bex (2017) to combine with the categorical features for the prediction of the online trade fraud complaints task (withdrawn or not withdrawn). By using Natural Language Processing (NLP) methods, Kos is able to correctly distinguish the withdrawn complaints from the non-withdrawn complaints with an average of 59% F1-score (macro averaged) on an imbalanced set. F1-score is an Accuracy measurement particularly emphasizing classification errors in imbalanced sets. When Hul combines the categorical features with the probabilities retrieved from the research of Kos, she observes a 1.38% increase in the F1-score (macro averaged) in comparison to the model that uses only the categorical variables.

Eirinaki, Vazirgiannis, and Varlamis (2003), create a recommender system solution called SEWeP (Semantic Enhancement for Web Personalization). In their recommender system, they combine the user log information with textual document clusters. They use the document clusters of the each web-page visited to extract the users' navigational behavior in the form of association rules.

To determine the effects of the combination of classical text features with non-classical text features, François and Miltsakaki (2012) apply linear and non-linear classification techniques to evaluate readability of the French texts. The classical text features include the number of the pronouns in one document and percentage of the words which are longer than 8 characters. On the other hand, the non-classical text features include the uni-gram probabilities and the cohesion measured via Latent Semantic Analysis (LSA). They report that the best readability performance is measured when they use the combinations of classical and non-classical text features.

In the domain of economics, Levenberg, Simpson, Roberts, and Gottlob (2013) combine textual information and time-series data streams to predict the trend of Nonfarm Payrolls (NFP) in the United States. NFP is an index that measures unemployment growth in the Unites States. It is used as an indicator of the US economic welfare. In their research, they first predict the sentiments for the daily news sentences and combine the predictions with four different economical indices, those include Consumer Price Index (CPI), Institute for Suppy Management Manufacturing Index (ISM), the JOLTS Nonfarm Index (JOLTS), and the Labor Force Levels (LFL). They report an increase in the accuracy with the complementary information of both textual information and economic indices.

Chapter 3

Dataset Description

The dataset that we use consists of 499 criminal case examples in which at the time of the crime the offender was between the ages 18 and 22 (both ages included). The criminal cases cover the time period in which the last offense related to the case is recorded between 5th April 2014 and 6th April 2018. Therefore, all the criminal case examples used in this thesis were applicable to the new revision on the juvenile justice system named as Adolescent Criminal Law (art. 77c, Dutch Criminal Code). Among 499 criminal law cases 158 are classified as Juvenile Law, whereas 341 cases are classified as Adult Law. As the label distribution is imbalanced, we use sampling methodology in our experiments. This methodology is provided in detail in Section 4.4.1.

The 499 instances that we use in this thesis consist of two distinct partitions as structured relational database and unstructured data. Structured and unstructured data have different characteristics and retrieval methodologies. Structured dataset has been provided by the WODC, it has been used for the Adolescent Criminal Law research program. The unstructured dataset is obtained by scraping the publicly available data of the Dutch Judiciary (De Rechtspraak, 2014). Those are joined by using the European Case Law Identifier (ECLI) and case numbers (Parketnummers), as those are the unique keys to be found in each dataset.

As unstructured data contains more specific level of information, we generated the dataset by considering the judgment text, which is publicly available. Data generation involved keyword search and web scraping. Following the data generation, we labeled the dataset generated, as there is no available information about the Juvenile (J) or Adult Law (A) outcomes for each of the criminal case. We used the unstructured dataset for labeling the criminal case examples. As a consequence we used the latest available conclusion, to define the adolescent criminal case outcomes. This is also consistent with our data merging strategy.

The structured dataset provided involved many manual entries and aggregated columns. Therefore, we pre-processed the available columns in detail. Besides the available columns, the structured dataset contains a label for the expert system. While running the experiments we decided to leave the expert system label outside of

the dataset, but we used that column for setting a reference point for our experiments.

In this chapter we will first provide a detailed information on the dataset generation. Then we will present and analyze the unstructured data and its components. We will provide a detailed analysis on the labeling strategy. We will present the categories related to the structured data, discuss the pre-processing strategy in detail and analyze the expert system performance by setting reference point for the practical goal. Lastly, we will give details on our data merging strategy.

3.1 Dataset Generation

We used the RESTful Application Programming Interface (API) that the Dutch Judiciary (De Rechtspraak, 2014) to retrieve judgment text files. The API gives Extensible Markup Language (XML) files as judgment texts. By using API calls with a programming language we cannot obtain keyword search results, as the Dutch Judiciary does not openly provide that option in the API. To solve this problem and to generate a dataset matching with structured dataset, we used four steps, which are summarized in the Figure 3.1.

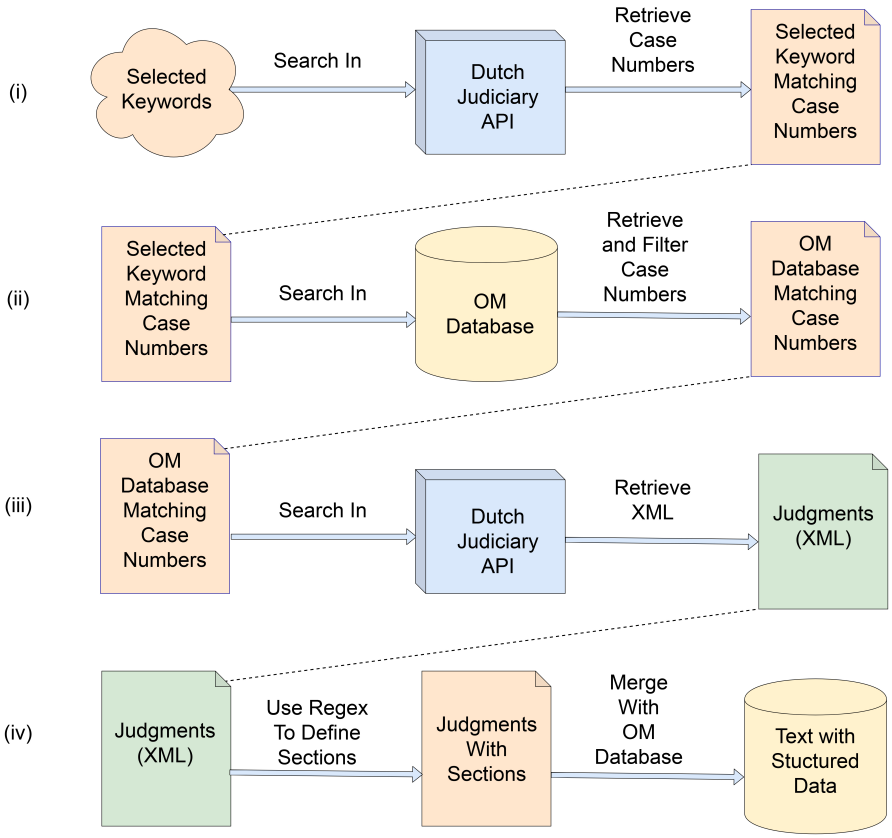


FIGURE 3.1: Dataset generation chart.

To start with, we used the Judicial Case Search Interface (De Rechtspraak, 1999)

to write a small web scraper that mimics internet users. The web scraper uses Selenium with Python to insert each of the keyword into the searching interface and track the keyword search results (Muthukadan, 2018). Selenium is a powerful tool to test or scrape websites that are using Javascript, a programming language for web to enable interactive interfaces. The web scraper collects the ECLIs that are matching with the initialized keywords (i).

To retrieve the relevant adolescent case judgment text files, we used related keyword queries. The domain experts in the WODC has been consulted for keyword selection. The following keywords are used for dataset generation:

- "adolescent" ("adolescent" in English)
- "geweld" ("violence" in English)
- "18 jaar" ("18 year" in English)
- "19 jaar" ("19 year" in English)
- "20 jaar" ("20 year" in English)
- "21 jaar" ("21 year" in English)
- "22 jaar" ("22 year" in English)
- "jeugddetentie" ("juvenile detention" in English)
- "minderjarig" ("under aged" in English)
- "ASR" (Dutch abbreviation of "adolescent criminal law" in English)
- "minderjarigenstrafrecht" ("under aged criminal law" in English)

For each of the keywords we selected the first 1,000 relevant (sorted by relevance, logic is provided by the Dutch Judiciary API) ECLI results. While scraping the ECLI results, we additionally retrieved case numbers. Due to the irregular entries of the judges, some of the ECLI and case numbers were not retrieved. Some judges were using different sections of the text documents to note down the procedural information. At the end of the step (i), we had 5,218 unique cases.

Not all the cases were contained in the structured dataset. Following the retrieval of the case numbers and ECLIs, we searched those in the structured database. We filtered the criminal law cases (with judgments), which have all commonly used sections. Then we matched the case numbers with the structured dataset. Lastly we filtered the criminal cases with an offender, who was between the ages 18 to 23 at the time of offense (ii).

At the end of the step (ii) we had only 499 cases remained for testing. Following that we used Dutch Judiciary API again to retrieve XML files (iii). Lastly we selected most commonly used sections with pattern matching to use as before trial section judicial text in our machine learning experiments (iv). When we obtained separated sections for judgments, we still had two separated datasets as unstructured and structured. To maintain consistency between both of them, we applied a merging strategy and stored our resulting dataset in a separate file.

Lastly, we annotated the case labels as either Juvenile Law (J) or Adult Law (A). The label annotation strategy is discussed in detail in Section 3.2.4.

3.2 Unstructured Data

The unstructured data is the judicial text and it is obtained from scraping the publicly available Dutch criminal law cases from the Dutch Judiciary. By using text tagging we divided each of the judicial text into sections. Following the scraping we pre-processed and analyzed the dataset with machine learning methods. The unstructured data is used for labeling the cases as Adult Law (A) or Juvenile Law (J) as well. This section will present the details on description of the data, pre-processing and analysis of the sectioned judicial text for each criminal law case. Lastly, we will provide the labeling strategy for training a predictive algorithm on the obtained dataset.

3.2.1 Data Description

Text data are extracted from the Dutch Judiciary Application Programming Interface (Rechtspraak API), by gathering the sentences with partitioning the sections. The API is public. It provides detailed judiciary sentences for a selected partition of the criminal cases by anonymizing the person related information. The Rechtspraak API does not provide a well-defined sectioning in the judicial sentences, as each of the local court uses different titles and sectioning strategies while notating. In order to define the sections properly, we first retrieved the section titles of all cases. We applied a frequency analysis on the section titles, to see which are the most commonly occurring ones. Then we combined the most frequent section titles with the same topic. After this analysis we made a decision on the grouping of the sections within the text. The sections are distinguished by using regular expressions, which is a specific text tagging method. The regular expressions that we used while sectioning the judicial text is provided in the Appendix A. We sectioned the judgment text into the following titles:

Juvenile Law	The suspect has tried to steal money from a prostitute under threat of a knife. Proof of attempted extortion. The court imposes a juvenile detention of 179 days with deduction of pre-trial detention and the PIJ measure conditionally with a probationary period of 2 years and, among other things, the conditions of clinical treatment and supervision of probation.
Adult Law	A 6-month prison sentence with deduction of the custody of a home burglary committed in association, followed by violence against reporters. In addition, the suspect must compensate the injured parties for the non-material damage of €200 and €300 respectively.

TABLE 3.1: Content Details examples translated to English.

1. **Content Details (Inhoudsindicatie):** In this section the summary of the evidences and the results are discussed within maximum 10 sentences. For some of the cases, judges also briefly mention that if the Article 77c. is applied as an

outcome of the case. Mostly, the key evidence and outcomes are represented in this short section. An example of Content Details section in the judgments of both J and A are given in the Table 3.1.

2. **Indictment (Tenlastelegging):** In this section the crime is briefly discussed and the procedural information on the case is given. This section can be seen as a summary section over the case, without giving any information on the sanctions that are considered within the case. It does not give a detailed information. Mostly, judges mention how they obtained the criminal case and what were the key evidence in the hearings. An example of Indictment section in the judgments of both J and A are given in the Table 3.2.

Juvenile Law	The accused is charged with the following: he on or around 15 January 2016 in Eindhoven for the execution of the crime intended by the accused to take away money, wholly or partly belonging to [victim] for the purpose of unlawful appropriation, in any case to someone else or others than the suspect, thereby precluding, intending to accompany and / or following violence the intended theft ...
Adult Law	The case was brought by summons on 13 February 2017. The accused was charged with the following: he was on or around 01 December 2016 in Oss, at least in the Netherlands, together and in association with one or more others, at least alone, with the intention of unlawful appropriation has taken away a jewelry box and / or jewelry and / or (foreign) money, in any case any good, wholly or partly belonging to [victim 1], in any case to someone else or others than the suspect and / or or his co-perpetrators ...

TABLE 3.2: Indictment examples translated to English.

3. **Evidence (Bewijs):** In this section the evidences are discussed and the assessment of the evidences are made. Evidence section can be seen as a textual aggregation of all the hearings, complaints of injured parties and witness statements. It contains very detailed information about the offence, including text messages (in case they contain evidence on an assault) and description of crime objects (in case the offense is related to hard drugs or weapons). An example of Evidence section in the judgments of both J and A are given in the Table 3.3.

   This section does not cover expert evidence or criminal law articles considered in the scope of the case. The Evidence Considerations cover, particularly, the following subtitles:

   (a) Evidence Considerations (Bewijsoverwegingen)

   (b) Evidence Statement (Bewezenverklaring)

   (c) Assessment of the Evidence (Beoordeling van het bewijs)

   (d) Evaluation of the Evidence (Waardering van het bewijs)

Juvenile Law	Based on the facts and circumstances contained in the evidence, the court considers that the accused: (primarily) on 15 January 2016 in Eindhoven for the execution of the crime intended by the accused for the purpose of being unlawful by favoring threats of violence by forcing [victim] to give up money belonging to [victim], said [victim] holding up a knife and saying "give me that money", or at least called "money", while the execution of that proposed crime has not been completed...
Adult Law	Regarding the alleged violence, he took the view that in the present case there is a distant connection between the use of force by the suspect and the theft. According to the lawyer, the theft and violence are spatially and temporally apart. He hereby refers to the words "followed by" and "when caught red-handed" in Article 312, first paragraph, of the Dutch Criminal Code. Since there was 1.5 to 2 hours between the theft in the home and the final arrest of the suspect and the arrest took place at a different location...

TABLE 3.3: Evidence examples translated to English.

4. **Punishment Motivations (Strafmotivering):** In this section the judges elaborate on the sanctions considered for the criminal case. Both the Prosecution Service and Court opinions about the punishments are mentioned in this section. It covers expert evidence or criminal law articles considered in the scope of the case.

Juvenile Law	The suspect has tried to steal money from the victim under the threat of a knife. The victim was in an extremely vulnerable position at the time of the robbery. She worked as a window prostitute and the defendant visited her as a customer. Only by the decisive action of the victim has the robbery not been completed. Through his actions, the accused has made a major violation of the privacy of the victim and has damaged her physical integrity...
Adult Law	A burglary in the home causes feelings of fear and insecurity among residents in particular and in society in general. In addition, a home burglary entails material damage and nuisance for the injured. The accused has not been concerned about all this. He was only guided by financial motives. The suspect and his co-perpetrators went to work in a planned way and traveled to Oss with the sole purpose of committing a burglary. In addition, the suspect used violence against reporters during his flight. Investigation officers deserve respect and respect from everyone. The court charges the suspect all of this...

TABLE 3.4: Punishment Motivations examples translated to English.

This section can be seen as a text, where all the evaluations about possible sanctions are made. An example of Punishment Motivations section in the judgments of both J and A are given in the Table 3.4. Punishment Motivations cover the following subtitles:

(a) Punishment Motivations (Strafmotivering)

(b) Punishment Considerations of the Facts (De strafbaarheid van het feit)

(c) Punishment Considerations of the Offender (De strafbaarheid van de verdachte)

(d) Imposition of the Punishments (Oplegging van straffen)

5. **Decision (Beslissing):** This section contains the punishments as a part of the decision of the case. It usually stands at the bottom of the judicial text and briefly mentions the decisions made over the case. The decisions contain the specific punishments in detail and articles considered and applied.

Juvenile Law	Youth detention for the duration of 179 days with deduction in accordance with Article 27. Placement in an institution for juveniles for a period of 3 years conditionally with a probationary period of 2 years.
Adult Law	Prison sentence for the duration of 6 months with deduction in accordance with Article 27 of the Criminal Code. Measure of compensation of EUR 200.00 or, alternatively, 4 days in custody. Therefore imposes on the suspect the obligation to pay to the State on behalf of the victim [officer 2] an amount of EUR 200.00 (in words: two hundred euros), in the absence of payment and redress replaced by 4 days in custody. The amount consists of an amount of EUR 200.00 for non-material compensation...

TABLE 3.5: Decision examples translated to English.

Besides of those sections indicated above, we retrieved Full Text contained in the judgement. Full Text corresponds to the full judgment, without filtering any specific section or title. The combination of Indictment, Evidence, Punishment and Decision sections in judgement text does not necessarily produce the Full Text. There might be some uncommon sections that the judges are using. Those uncommon sections are also contained in the Full Text retrieval. Additionally, when we say Full Text, we do not include the Content Details. In the API the Content Details are presented within a completely different element in the XML files. Thus, we treat those separately.

As we defined the sections above, the before trial sections are selected to be Indictment and Evidence. Because they do not contain any information about sanctions or article applications considered. Those two sections are used in the experiments for the research questions stated in Section 1.1. To improve the accuracy with as much information as possible we use Content Details and Full Text in the experiments for the practical questions stated in Section 1.2.

3.2.2 Pre-processing of the text data

Following the extraction of the text information for both classes, *tokenization* for each of the section partitions for the corresponding cases will be implemented to convert the raw text into character sequences. Also, the upper case characters will be transformed into lower cases to unify the structure of the tokens. Right after this step, transformations for the term extraction will be applied. Those will

include stop-word removal, date tagging, lemmatization, term frequency and inverse document frequency matrix and N-grams.

- **Stop-word removal:** Stop-words are the words that do not provide much information in context, such as "a", "an" and "the". Typically, a pre-defined set of commonly occurring words in a language is used for the retrieval process. In this study, we use a pre-defined stopword file, provided in a publicly available repository (Ander, 2019).

- **Lemmatization:** While processing a document, two similar words can be tokenized distinctly. This can create noise while applying an automatized method for analysis. However, there are two ways to solve this problem, which are Stemming and Lemmatization. Stemming unifies the words sharing the same root whereas Lemmatization unifies the words sharing the same meaning. Regarding Dutch language, Stemming is prone to stemming mistakes. For instance, the word for the former monetary unit of the Netherlands *gulden* ends with -den, which can be confused with the past tense suffix. In that case, the Stemmer may reduce the word "gulden" into "gul" which is a completely different word with the meaning of "generous" (Gaustad, 2004). To prevent such errors, we will use Frog (with Python) for Lemmatization to process Dutch documents (Bosch, Busser, Canisius, & Daelemans, 2007).

- **Date and Money amount tagging:** A judicial text often contains important dates and the amount of monetary punishment or of bribery. As we will follow a BOW approach, some of the day or year indicators might lose their meanings. We will tag each of the numerical day, textual month and numerical year combination as a "__DATE__" to avoid such confusion. We will also tag the monetary indications as "__MONEY__". By implementing these additional steps, we aim to remove some of the noise that could occur due to the numerical characters in text.

- **Term Frequency and Inverse Document Frequency Matrix:** A common way to come up with a numerical representation of the words appearing in each document is to calculate the term frequencies, *tf*. This will result in a non-negative and sparse matrix with a dimension of $n \times d$. We will additionally use inverse document frequencies (*idf*) to reduce the impact of frequently occurring terms in the document collection. This step will be implemented using the scikit-learn package of Python programming language (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, et al., 2011a).

- **N-grams:** The-Bag-of-Words (BOW) is a commonly used approach in the literature of Natural Language Processing (Salton & McGill, 1986). In the BOW approach, a document is represented as a set of words (unigrams) or a

sequential set of multiple words (N-grams). As N-grams are reported to be more effective in the text classification tasks, we will apply 1-to-2 N-grams and retrieve the top 4,000 frequent N-grams while constructing the tf-idf matrix.

3.2.3 Data Analysis

To have insights about our text dataset, we have implemented a few exploratory analysis methods for each of the sections separately.

First of all we tried to measure the similarities between the sections that we retrieved. The similarity scores are obtained using the same Text pre-processing strategy that we use for the classification tasks. We used 1 to 2 N-Grams, a fixed set of Dutch stop words and inverse document frequency to obtain a tf-idf matrix. The dictionary is kept the same for each of the sections. That is to say, we fit the tf-idf scores by using the whole combined text for every section retrieved. Then we calculated the tf-idf for each of the section separately. After pre-processing, for each of the section combinations we applied row by row (case by case) cosine similarity distance calculation and take the mean of the resulting array. The Figure 3.2 shows the relationship between the combinations for each section.

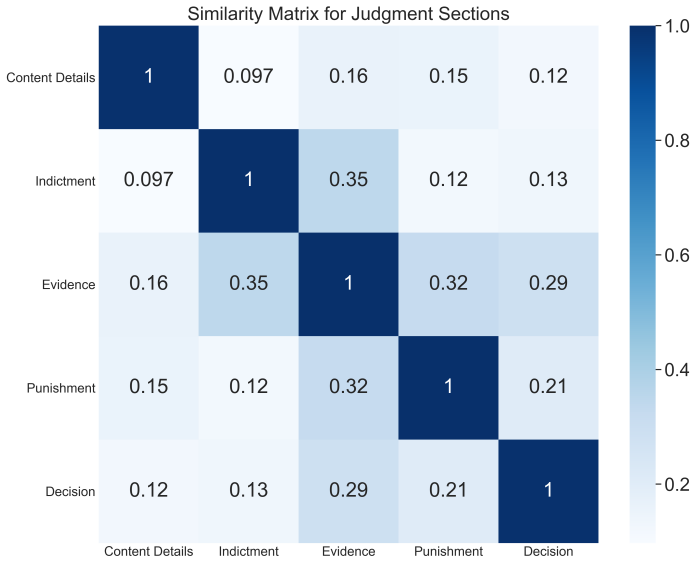
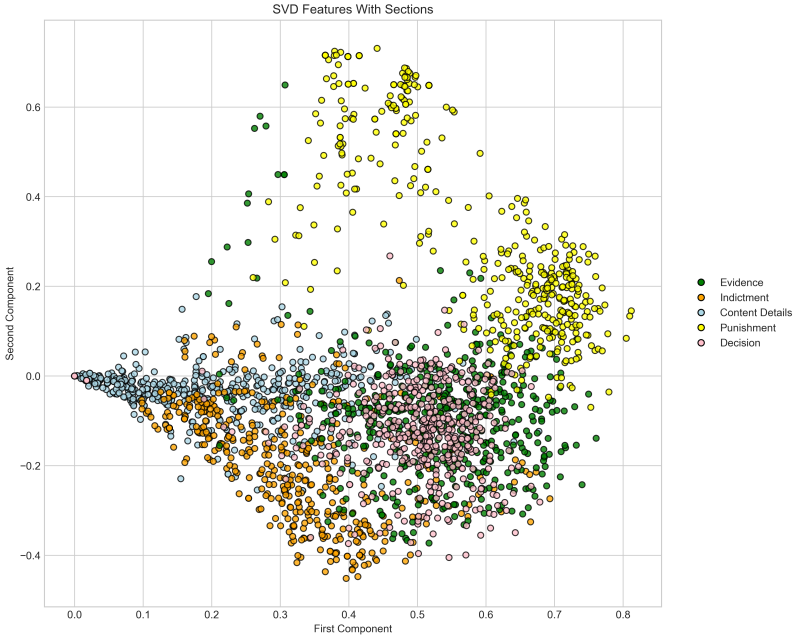


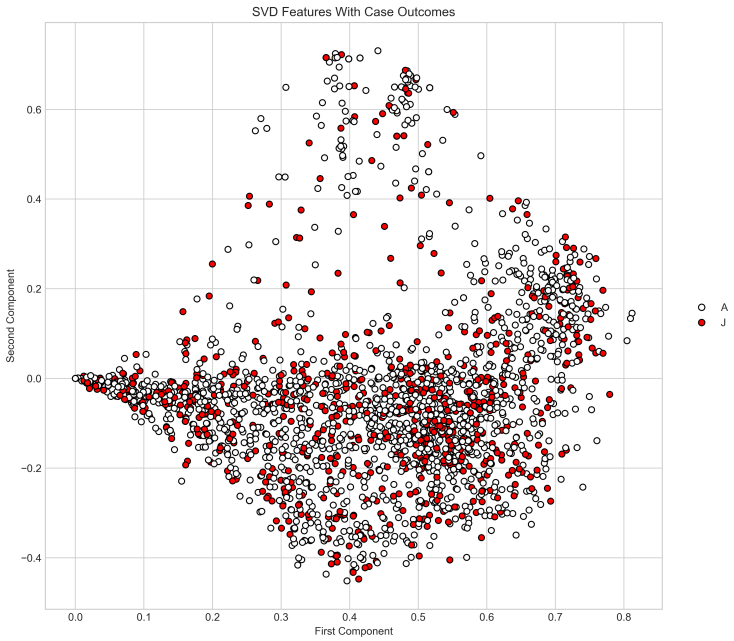
FIGURE 3.2: Similarity matrix for each of the sections.

There is no high cosine similarity found between the sections. On the other hand, the resulting figure implies that Evidence Considerations have more similarities with other sections. Besides that, Evidence and Indictment are the most similar sections in comparison to the others. Interestingly, Punishment section has some similarities with the Evidence Considerations. These results may give some hints as the Evidence

Considerations section has some common occurrences with the Punishment and Decision sections. Therefore, according to the analysis Evidence Considerations contain more information related to the outcome of the Adolescent Law cases dataset that we have.



(A) Sections



(B) Labels (A or J)

FIGURE 3.4: Truncated SVD plot for sections and labels.

Secondly we tried to visualize each section by reducing the dimensions of those.

The dimension reduction is done by using Truncated Singular Value Decomposition (Truncated SVD) as this method reveals the similarities between the instances (Husbands, Simon, & Ding, 2001). The Truncated SVD is calculated by using a tf-idf matrix using a union of the vocabularies used in all the judgment text sections. The tf-idf matrix is reduced into 2 dimensions (projected documents) to visually analyze different sections.

Figure 3.3a shows the 2-dimension mapping of the reduced dimensions for each of the sections. By looking at that we can again imply that Evidence Considerations section (Green) have lots of commonalities with again the Decision section (Pink).

While analyzing the sections the top of the plot, which represent the Punishment section (Yellow), two distinct clusters were revealed. Therefore, we wanted to know whether those clusters reveal the labels (Adult Law or Juvenile Law). Figure 3.3b shows that labels are distributed homogeneously among the Truncated SVD reduced representation of the unstructured dataset.

3.2.4 Label Annotation

The labels of the dataset indicate Adult or Juvenile Criminal Law decision of the judge. They are notated by the author of this thesis, who does not know Dutch. She used Google Translate to manually distinguish between the decisions. The accuracy of the non-Dutch speaker has been measured by randomly sampling 30 instances from the 499 cases selected. The random selection and validation is done by a Dutch speaker domain expert. The labeling accuracy is found out to be 100% for the selected cases.

The notations are based on the Dutch Criminal Law articles and sanctions considered in the judgment. The article numbers between 71 to 79 indicate the Juvenile Law decisions. Other articles (except for the art. 71 to 79 Criminal Code) indicate Adult Law decisions.

As an exception to the regular Adolescent Law application indications, there were some case decisions that indicate both Juvenile and Adult sanctions at the same time. For instance some case decisions indicate the combination of juvenile detention and imprisonment or combination of PIJ measurement and probation. In those exceptional cases, we assumed those to be Juvenile Law decisions. To conclude, if we observed any indication of any Juvenile law articles or measurements, the case is labeled as Juvenile Law and we considered that art. 77c is applied.

3.3 Structured Data

The structured dataset has been retrieved from OBJD (Onderzoek- en Beleidsdatabase Justitiële Documentatie)[1] and it is used for Adolescent Criminal Law and Recidivism research in the WODC (Wartna, Blom, & Tollenaar, 2011). In the structured dataset, there is also an expert system that defines if the case belongs to Juvenile Law (J) or Adult Law (A). The following will first give details on the shape and columns in the dataset. Then we will give information on the pre-processing of the structured dataset. Lastly, we will compare the dataset labels with expert system labels.

3.3.1 Data Description

The dataset is manually aggregated by using the Integrated Criminal Justice Process System (Geïntegreerd Processysteem Strafrecht) and it includes 373s attributes, except for the id columns. The types of the structured information are grouped as follows:

1. **Sanctions:** The relevant measurements, punishments and decisions of the case. It includes the financial, freedom related punishments and preventive measurements as the decision of the case. In the categorical variables, there is also a distinction between the juvenile and adult type of measurements. For instance, the juvenile detention (jeugddetentie) is a different variable in comparison to the imprisonment.

2. **Procedural information:** The relevant procedural information of the case. It includes the judge type, the place of the court and date related information of the trial. It does not include any information about the decision.

3. **Criminal information:** The relevant criminal information of the case. It includes the offense types and the prior criminal history of the offender. It does not include the actions after the beginning of the prosecution.

4. **Personal information:** The relevant personal information of the offender prosecuted. It includes the birth date, birth place information of the offender. The variable count in this category is relatively smaller than the other categories.

The distribution of the groups are shown in the Table 3.6. There were some variables that are arguable to be in the procedural types or in the sanction types. Especially, those two types were hard to classify, since there is no clearly defined semantics information for the structured dataset. There are some variables that are

[1] As the structured dataset originates from Dutch Public Prosecution Service (Openbare Ministerie), we call it shortly as *OM Database* in this thesis.

Type	Variables
Criminal	89
Personal	7
Procedural	66
Sanctions	211
Total Number	373

TABLE 3.6: Database categories.

using the phrase "the number of times" in their title name for defining a type of measurement or word is indicated in the case. For instance, there is a variable saying "Number of times monetary measurement" (Aantal keer geldstraf), we classified that as Punishment. Only if there is no sanction indication in a description of a variable, we classified those as Procedural.

We used almost the same current classification that Dutch Public Prosecution Service uses. However, we considered those to be following a time frame, from the starting from the birth of the offender till the case decision. The structure is shown in the Figure 3.5. We would expect using only the Sanctions data to give higher accuracy than using only the Personal data of the offender.

FIGURE 3.5: Structured database information in a time frame.

3.3.2 Data Pre-processing

The structured dataset contains many aggregated columns, which are not clear even for the domain experts. Pre-processing is needed. Therefore, while cleaning the structured dataset we consider the following rules:

- **Categorical Variables:** We will get dummy variables for those, and remove the first category.
- **Missing Values:** Those are the non-existing values in columns with different data types. Therefore, while interpolating those we need to treat each data type separately.
  - We fill the missing values in the numerical columns with their mean value.
  - We fill the missing values in the dummy variable columns with 0.
  - We fill the missing values in the date columns with an fixed initial date (1st of January 2014).

- **Yes or No Variables (Binary Variables):** If the given measurement or the procedural situation is included in the case those columns are valued as "Yes", otherwise as "No". The "Yes" values are transformed into 1, and the "No" values are transformed into 0.
- **Dutch Criminal Law Article Variables:** Those indicate the related article numbers according to the Dutch Criminal Law, those are manually inserted fields, which contain character based articles. We count the number of articles considered per case and create a new column for that. We will delete the article variables as they include only text.
- **Criminal Activity Variables:** There are additionally some columns that indicate criminal activity in a case. Those criminal activity columns contain categories of crimes with string values separated by commas. Those are transformed into dummy variables.

After the pre-processing of the structure database, we expect to have a dataset without any missing values. Additionally, we expect to fill the missing values and the feature engineering in such a way that adds value to our research.

3.3.3 Comparison of the Labels with the Expert System

As stated in the Section 1, structured database uses an expert system to insert the labels by checking the sanctions. Expert systems defined as "computer programs, which involves non-algorithmic expertise for solving specific problems". For a particular domain, domain experts' knowledge is encoded into an algorithmic form, often represented in IF-THEN rules (Meritt, 1989). In the OM dataset, there is also an integrated expert system which uses logical rules to label the criminal law cases as A or J.

After labeling unstructured data, we compared our results to understand the current situation of the expert system and have a reference point quality measurements. Before analyzing the expert system, it is useful to review its pseudo-code.

The pseudo-code is shared in the Algorithm 1. The algorithm basically checks the Sanction columns (in the same OM dataset) with the AND and OR logical statements. The Sanction columns contain dummy variables, stating whether or not the given sanctions are considered for the corresponding case. There are two paths that the expert system follows:

- **Juvenile Path:** In the Juvenile Path, the expert system checks whether one of the sanctions listed is indicated in the case text: reprimand, legally under surveillance, juvenile provision or placement, arrest indication, reform school or behavioral sanctions for juveniles. Then the outcome of OR statement is assigned to a variable called "not adult". This is further used in the algorithm

Algorithm 1 Expert System

1. if
   (reprimand or
   legally under surveillance or
   placement in juvenile facility or
   youth-provision with medical case or
   arrest (arrest) or
   reform school (tuchtschool) or
   behavioral measures for juveniles) then not adult;
2. if
   (imprisonment (gevangenisstraf) or
   detention (hechtenis) or
   placement in systematic offenders institution or
   military detention or
   placement in psychiatric hospital or
   public service or
   involvement of SOV (Criminal Addiction Ministration) or
   provision of psychiatric disorder (Terbeschikkingstelling) or
   provision with health care (Terbeschikkingstelling) or
   provision of conditions (Terbeschikkingstelling) or
   provision without health care (Terbeschikkingstelling)) then adult;
3. if not adult == 1 and adult == 0 then "Juvenile Law";
4. if not adult == 0 and adult == 1 then "Adult Law";
5. if type of judge == "Children Judge" then "Juvenile Law";
6. if indication of juvenile law == 1 then "Juvenile Law";
7. if indication of juvenile law == 0 then "Adult Law";

to compare the outcomes with another variable "adult", that comes from the Adult Path.

- **Adult Path:** In the Adult Path, the expert system checks whether one of the sanctions listed is indicated in the case text: imprisonment, detention, systemic offenders institution, military detention, placement in a psychiatric hospital, public service, criminal addiction or one of the possible provisions[2]. Then the outcome of OR statement is assigned to a variable called "adult". This is further used in the algorithm to compare the outcomes with another variable "not adult", that comes from the Juvenile Path.

After following IF statements in the lines 1 and 2, the lines 3 and 4 checks the mutually exclusiveness of both "not adult" and "adult" variables. If one or more of the "not adult" sanctions and none of the "adult" sanctions are considered in the

[2]Terbeschikkingstelling (TBS) is a sanction in the Dutch Criminal Law, stating the provision by the government. Judges impose TBS, if they emphasize the treatment of an offender.

case, then the expert system predicts the outcome as Juvenile Law. Conversely, if one or more of the "adult" measurements and none of the "not adult" sanctions are considered in the case, then the expert system predicts the outcome as Adult Law.

Confusion Matrix	True Labels	
Expert System Labels	Adult	Juvenile
Adult	157	90
Juvenile	1	68

TABLE 3.7: Confusion Matrix for the expert system that OM uses.

There are two special cases in the algorithm. One of them takes the implication of Juvenile Judge (Kinderrechter) into account in the case text. Then the algorithm predicts the outcome as Juvenile Law (line 5). In the second case the algorithm checks for an indication of the Juvenile Law (lines 6 and 7). If there is an indication, then it predicts as J. Else, it predicts as A.

Although they are considered in a case, the OM structured database system does not indicate some of the measurements. Also, there are many missing values due to manual entry methodology that they use. Therefore, the sanction columns also contain many missing values. Besides that, expert system labels are also missing for same case observations. While providing a basic analysis, we will consider those missing expert system labels as A.

We used the labeled dataset that we currently use for our experiments to calculate a confusion matrix. Before creating the confusion matrix, we first balanced the dataset by randomly sampling the Adult Law (A) decisions as they are the majority class. We used the same random seed as our experiment random seed to be consistent.

Table 3.7 shows the confusion matrix, in case that is used for making the predictions for distinguishing the Adult Law (A) decisions versus the Juvenile Law (J) decisions. The expert system is able to classify all the Adult Law decisions correctly, whereas it is not able to predict more than half of the Juvenile Law (J) decisions.

By using the confusion matrix above, we can calculate reference point quality measurements for our experiments while answering the practical questions as both use after trial information of the cases. We obtained 70.5% Accuracy and 58.2% F1-score. See Subsection 4.4.3 for more details on quality measurements.

3.4 Merging Unstructured and Structured Datasets

As briefly mentioned in Section 3.1, we wanted to reduce the inconsistencies while using both unstructured and structured data. To do that we analyzed the identification variables, ECLI and case numbers, and followed an efficient merging strategy.

ECLI is an identifier for case law that contains the metadata about the updated case conclusions (Opijnen, 2011). For each updated conclusion, one case obtains an additional judgment text, labeled by an ECLI. Hence, with each updated judgment the number of ECLIs connected to a particular case increases. Similar to the ECLIs, case numbers indicate metadata information about the cases. However, they include the aggregated conclusions or updates for a given case.

Since the unstructured data is a judgment text file with an updated conclusion, it contains ECLI level of information. Hence, if we know each ECLI for a given case, we can reach all the updated conclusions for that particular criminal case. On the other hand, the structured data contains case number level of information about the trial procedures, sanctions, crimes related to the case and the offender's personal information. Thus, the structured data contains a general level of information, whereas the unstructured data contains more specific information related to the case.

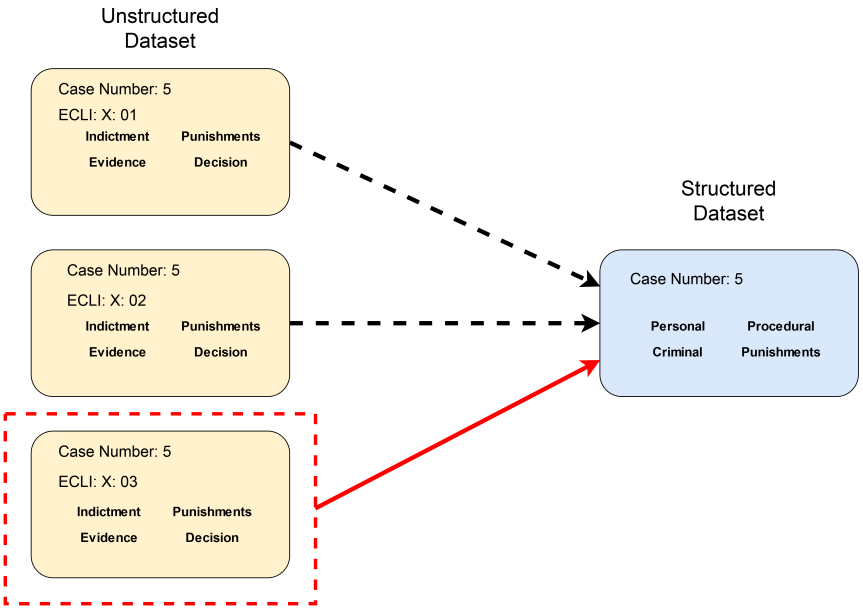


FIGURE 3.6: The solution for the one-to-many relation of the Structured and the Unstructured dataset.

As structured data can be connected to many judgment text files, this would lead to a one-to-many relationship. Therefore structural conflicts might emerge as we run machine learning experiments. For instance, we might have structured information relevant to *Case Number: 5*. That information might correspond to one row in the structured database. However, the *Case Number: 5* might have been updated three times, resulting with three different ECLIs and judgment text files (*ECLI:X:01*, *ECLI:X:02*, *ECLI:X:03*).

Figure 3.6 shows the solution that we developed to tackle that structural problem. While merging the unstructured and structured data we selected the latest ECLI-conclusion for the corresponding case number to eliminate the one-to-many relationship. Hence, the latest available judgment text is joined with the structured

dataset. The older versions of the case decisions are dropped.

Chapter 4

Methodology

The present study will use both database information (structured) and textual data (unstructured) to predict the adolescent criminal case outcomes. Text data is often created by human participants, and therefore it may contain misspellings, ambiguous words, and highly varying grammatical usage. That is why different methods of data transformation will be utilized in the pre-processing of the unstructured data.

Three selected classification algorithms will be applied to both data types. For the text data, the machine learning approach will be tested with and without dimension reduction techniques.

After testing the text information and the database information separately, we combine the pre-processed text with the database information via machine learning pipelines. The resulting pipeline will end with a classification algorithm to run our experiments. Figure 4.1 shows a general overview of the machine learning pipeline to be implemented. To answer the research questions, we also need to implement some steps optionally. For instance, while running experiments, we should both include the reduced dimensions for text and the full term frequency and inverse document frequency (tf-idf) matrix to implement classification. We should also run the experiments with text-only and database-only features before combining them. A detailed explanation of the optional paths to be implemented can be seen below:

- Path 1: This path indicates the processing of the text features. We will first apply the pre-processing techniques unconditionally. Then, there will be an option (Option DR) to allow for applying DR techniques that are mentioned in the Section 4.1.
- Path 2: This path indicates the processing of the structured database features. To implement this path, we will clean the structured features unconditionally. There will not be any step taken to reduce the dimensions, as none of the research questions require such action.
- Path 3: This path indicates the combination of Path 1 and Path 2. To implement this path, we will use both Path 1 and Path 2 features both with Option

DR and without Option DR. In our experiments we include the results obtained both with Option DR and without Option DR while implementing Path 1 and Path 3.

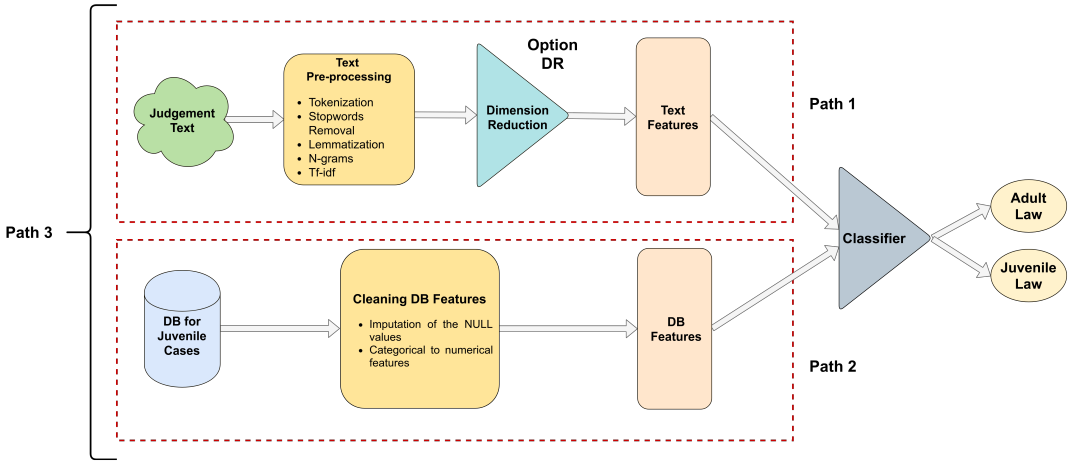


FIGURE 4.1: The experimental setting of the study.

Since each of the different data types have a different scale and the selected classification algorithms are sensitive to differently scaled data at the end of the data pipelines (before classification) we apply standardization. For instance, in the experiments we apply Logistic Regression with L_2 penalization (i.e., Ridge Regression). While applying Ridge Regression, the penalization is determined by the variances of each variable. If one of the variables has a scale between 0 to 1000 and another variable is a binary variable with a value of either 0 or 1, the coefficients can be wrongly determined. One can tackle the problem by *centering* the data (Hastie, Tibshirani, & Friedman, 2017c). Hence, for each of the Paths defined (Path 1, 2 and 3) we apply the standardization defined as in Equation 4.1 before applying each of the selected classification algorithms. For all the independent variables the average value will be subtracted from the value and the resulting centered data is divided by the standard deviation.

$$z_i = (X_i - \mu)/\sigma \tag{4.1}$$

The following Sections provide more detailed information on the pre-processing of the text data, the applied DR algorithms, the combination mechanism, applied classification algorithms and quality measurements.

4.1 Dimension Reduction Algorithms

Text information often contains noisy information with a high-dimensional representation. Dimension reduction (DR) techniques combine the high dimensional features to construct a more compact representation and therefore reduce noise. Since

they provide a compact representation, they are quite useful to solve the problems of the synonymy and polysemy within the text (Aggarwal & Zhai, 2012c). Due to the computational limitations of this study, only the most common DR techniques are investigated. These include Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) and Word2Vec.

Latent Semantic Analysis is a matrix factorization technique used for creating conceptual representations of word-document matrices in texts. This method uses Singular Value Decomposition (SVD) as a matrix factorization technique to obtain the semantic concepts inside the text documents (Landauer & Dumais, 1997). SVD constructs an arbitrary rectangular matrix of terms and documents. By decomposing this rectangular matrix into three matrices SVD shows the breakdown of the original relationships between features into linearly independent components. By truncating the small components, LSA approximates a feature matrix with fewer dimensions (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). This technique generates a general representation of the corpus by using local co-occurrences (Landauer, Foltz, & Laham, 1998). Thus it is likely to capture some aspects of synonymy and polysemy.

As an alternative to LSA, one can approach the problem in a probabilistic way. Probabilistic Latent Semantic Analysis (pLSA), was a significant step for having a statistical latent class model (Hofmann, 1999). This approach describes each text document as a sample of mixture components (i.e., representations of topics). However, both LSA and pLSA were based on the assumption of "bag-of-words" that the order of words in a document might be neglected (Blei, Ng, & Jordan, 2003). On the other hand, two bottlenecks were reported in the algorithm: the increase of the parameter space that grows linearly with the size of the corpus and being not clear while generating probabilities for out-of-sample examples. With the assumption of exchangeability (i.e, specific ordering of the documents in a corpus can also be neglected) Latent Dirichlet Allocation provides a generative and efficient solution while addressing two bottlenecks of pLSA (Aggarwal & Zhai, 2012c).

Latent Dirichlet Allocation (LDA) is a topic modeling technique that learns the probabilities of the occurrences of the terms in a topic learned by the same algorithm (Aggarwal & Zhai, 2012c). The algorithm uses Dirichlet distribution to generate the topics of one document as a one-shot process (Aggarwal, 2018b). Therefore, the document specific parameters are generated by a compact set of parameters (in the Dirichlet distribution). This reduces the parameter space in comparison to pLSA and reduces the overfitting (Blei et al., 2003). As an output of the algorithm, the topics and the related words are explicated.

Besides of the matrix factorization and topic modeling approaches for text, word embeddings can be used for reducing dimensions as well. Embeddings map the words, sentences or documents to numerical values or vectors. Word2Vec is one of the most

popular and efficient embedding methods that are available in the commonly used machine learning programming languages like Python. Word2Vec is a word embedding method and tries to maximize the accuracy of the vector representations while preserving the linear regularities between the words (Mikolov, Chen, Corrado, & Dean, 2013). Word2Vec is accepted as an additive model, as a result, adding two vectors captures the meaning of the closest word vector (Gittens, Achlioptas, & Mahoney, 2017). A famous example would be adding *man* and *royal* to obtain *king*.

By training a Word2Vec model on a large Dutch judgment corpus we can obtain word representations relevant to the legal domain. As our main problem in this thesis deals with the criminal law case documents as a whole, we need to have an aggregate representation of the words used within a judgment. In the text mining in legal domain literature Mandal et al. (2017) aggregate the word embeddings by weight averaging with tf-idf scores for the corresponding term frequency in a given document. Since Word2Vec is accepted as an additive model, they expect their approach to benefit from it. We will, therefore, adapt their approach in our experiments by summing (with weights) the word vectors row wise.

In the experiments we use a Word2Vec with 200 dimensions, which is trained on over 2 million available judgments (without lemmatization) in the Dutch Judiciary API (Narayan, 2019). We retrieve the word vectors for each of the words used in the given documents. Consequently, we retrieve tf-idf scores for each word (1-gram) in the given documents. We weight sum row wise the word vectors by using the 1-gram tf-idf scores. In the end of this DR operation we obtain 200 dimensions of weighted sum word vectors for judgment documents. Equation 4.2 shows the weighted sum approach that we use in this study.

$$\sum_{i=0}^{k} \text{tf-idf}_i \cdot \text{Word2Vec}_i \tag{4.2}$$

where: k = count of the words in a text document d_i

The co-occurrences and the variation of the topics in the texts depend on the corpus that is used. Moreover, it is hard to come up with a pre-defined number of dimensions to explain the semantics needed for a specific classification task. As a result, we optimize the number of the dimensions for each of the DR algorithms by the experiments. This process will be done as a hyper-parameter tuning step, except for the experiments involve Word2Vec. The text parameter settings are summarized in Table 4.1.

Setting Name	Parameters	
N-Grams	1-2	1
Top Freq	4,000	None
Dimension Reduction on Text	[None, LSA, LDA]	[Word2Vec]
Number of Dimensions	[20, 40, 60, 80, 100]	200

TABLE 4.1: Text pre-processing settings for all classification algorithms.

4.2 Combination of Unstructured and Structured Data

The present study requires a reproducible way of combining unstructured and structured information retrieved related to the Dutch Adolescent Law (article 77c, Dutch Criminal Law). It is a challenging task, as it involves both DR and feature selection steps. This is possible by using a machine learning pipeline. A proper machine learning pipeline connects a series of data transformers together in a meaningful way to construct a single unit chain. The outcome of this applied method could be used for data extraction from various sources, the transformations of the extracted data, normalization the transformed data, and training a machine learning model (Buitinck et al., 2013). This approach will enable the machine learning experiment setting to be consistent and simplified while performing complex and nested workflows, particularly when working with text transformers (Stewart, 2014; Ioannides, 2017). For constructing a machine learning pipeline, Feature Union and Pipeline objects from the scikit-learn package of Python programming language will be used (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, et al., 2011a).

4.3 Classification Algorithms

The problem of classification will be formally defined as follows. Each of the judicial sentences and their combination with the structured information will be defined as a *document*. Although we name it similarly to the *documents* in text-only classification tasks, this time we will contain the structured information in the *documents* as well. As the structured information will contribute to the numerical matrix of the text documents, we will treat those similarly. The case classes (whether the sentences are under the common law or juvenile law) will be the set of predefined categories. Adopting the text classification notation described by Sebastiani (2002), we will have a task of assigning a Boolean value to each pair of documents and classes as $\langle d_j, c_i \rangle \in D \times C$, where D is a domain of documents (judicial case text and their combinations) and $C = \{c_1, c_2\}$ is the domain of classes defined as the applied law category (common law or juvenile law). In the automated text classification task, we

will try to find the best-approximated *classifier*, that matches $\langle d_j, c_i \rangle$ with the right classification decisions.

We are unable to implement every classification algorithm, and test it as there are many classification algorithms and different settings for each of them. To start with we investigate the most common classification algorithms for document classification. Text data have high-dimensions and sparsity as the main characteristics. Thus, linear models are particularly suitable for the text domain (Aggarwal, 2018c). In terms of linear models we have chosen two most common ones: Logistic Regression and Linear Support Vector Machine (SVM). As the kernel setting in SVMs allows us to use SVM as a non-linear classification method as well. However, since we apply multiple classifiers with different hype-parameters non-linear kernel for SVM is thought to be inefficient in terms of running time. Besides those two linear classification algorithms, we will investigate a more efficient boosting method (Light Gradient Boosting) that uses Decision Trees as base estimator.

Logistic Regression is a member of the Generalized Linear Model family, which are adapted to deal with discrete dependent variables, e.g., binary, ordinal, categorical and count data. It is a probabilistic model, and uses maximum likelihood estimation to fit (McCullagh & Nelder, 1989). The Logistic Regression tries to model the subsequent probabilities of K classes (in our problem it would be a binary class as c_1 and c_2) with linear functions in the predictors, x (Hastie, Tibshirani, & Friedman, 2017b). Equation 4.3 shows the standard Logistic Regression model.

$$Pr(X_i) = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)} \tag{4.3}$$

In the Logistic Regression, *Regularization*, imposing a penalty on the size of regression coefficients, is a frequently used way to eliminate the noise resulting with high dimensional data. Regularization in Logistic Regression is controlled by a parameter λ. Larger λ indicates a greater penalty. There are two common ways to impose Regularization into a regression problem as L_1 (Lasso) and L_2 (Ridge). Both L_1 and L_2 are expected to shrink the estimates of coefficients towards zero. With L_2 the shrinkage will be towards zero, whereas L_1 will be forcing some of the coefficient estimates to be exactly equal to zero and perform an additional *variable selection* step (James, Witten, Hastie, & Tibshirani, 2014).

Linear Support Vector Machine (SVM) is a classification algorithm that enables Regularization as well. The main idea in SVM is that it creates a hyper-plane or multiple hyper-planes to separate the training observations into the two classes. It maximizes the margins (distance) between the observations of two classes.

Equation 4.4 shows the standard hinge-loss minimization problem of Linear SVMs, where J is the loss function. $\frac{1}{2}\|\overline{W}\|^2$ stands for the margin between two hyper-planes

[1] and $\sum_i \max(0, 1-y_i[\overline{W}\cdot\overline{X_i}])$ for the margin violation. While maximizing the margin according to the hyper-plane, SVM allows us to impose a cost parameter, C. This parameter helps us to discourage margin violation and over-fitting (Aggarwal, 2018a). If the C is small then the SVM will increase the Regularization, while widening the margin.

$$minimize\ J = \frac{1}{2}\|\overline{W}\|^2 + C \cdot \sum_i \max(0, 1 - y_i[\overline{W} \cdot \overline{X_i}]) \tag{4.4}$$

Decision Trees provide a proper structure for the context of text data, as a hierarchical decomposition of the presence or absence of one or more words in the document is typical (Aggarwal & Zhai, 2012a). However, decision trees are generally prone to overfitting, as they can create over-complex trees. By combining decision trees we can have less overfitting and stronger (by means of predictive power) classification and regression algorithms. In the machine learning literature, the method of combining multiple predictive algorithms is called *ensemble learning*.

There are two popular ensemble techniques that initially use Decision Trees as a weak classifier: Bagging and Boosting. *Bagging* is an ensemble method, where a committee of Decision Trees produce a vote of outcome to be aggregated to result as a prediction. On the other hand, *Boosting* algorithms use a meta-algorithm based on a committee of Decision Trees to iteratively improve the performance by updating the meta-algorithm dependent on the results from the previously built models within the training phase of the algorithm (Hastie, Tibshirani, & Friedman, 2017a).

The main idea in boosting is to iteratively combine the weak classifiers to produce a more powerful ensemble by in each step reducing the fitting errors. Gradient Boosting is a boosting mechanism that initially uses Decision Trees as base classifier. And there are a few implementations in the programming packages, including XGBoost (Chen & Guestrin, 2016), sci-kit learn (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, et al., 2011b) and R (Ridgeway, 2007). Light GBM is a recent implementation of Gradient Boosting. It proposes an efficient algorithm to deal with large number of instances and features. In their paper Ke et al. (2017) claim that Light GBM outperforms the most popular Gradient Boosting algorithms in terms of memory consumption and computational speed, as it can have similar performance. As we have a large number of feature sets (text data and structured data), to obtain efficient experiment results with a Boosting algorithm (using Decision Trees as base classifier) we choose to use Light GBM.

[1] Margin between two hyper-planes is noted as $2/\|\overline{W}\|$. However Equation 4.4 contains the squared inverse of this quantity. Therefore minimizing this would be equivalent to maximize the distance between two hyper-planes.

4.4 Classification Evaluation

While evaluating the results we use a balanced set of criminal law cases and a repetitive approach of validation. We keep the hyper-parameter set of each machine learning algorithm used in this thesis constant while evaluating the performance. Besides we use relevant quality measurements. In the following sub-sections the details of our evaluation approach will be given.

4.4.1 Sampling and Validation Methods

The dataset that we use in this thesis is balanced as we aim to under-sample the Adult Law cases as the same number of Juvenile Law cases for the experiments. While answering the research questions by using classification validation methods, we also want to keep the number of the Juvenile and Adult Law cases same, as our research questions only the impact of DR and combination of different characteristics of datasets.

158 of the labels were classified as Juvenile Law and the rest as Adult Law. As the initial distribution of the labels is 70% to 30%, in the experiments we randomly under-sampled the Adult Law cases to obtain a balanced distribution of the labels. Hence, in each experiment we use 316 adolescent criminal law instances with a balanced label (50% and 50%) distribution. The random selection is done by using a fixed random seed for the each experiment that we run.

To identify a proper validation method, we first search the machine learning applications in the law domain. A majority of the predictive analytics research on the legal domain, which are explained in the Section 2.1.2, use Stratified K-fold Cross Validation (CV) to estimate and average quality measurement. For optimizing the classification algorithm settings they use an additional step, we call that step as hyper-parameter tuning. We give more details about the hyper-parameter tuning step in the Subsection 4.4.2.

K-fold CV splits the dataset into K mutually exclusive subsets (the folds) with approximately equal sizes. For each of the iterations one of the folds is kept as test set and the rest are kept as training. The training fold is used for fitting the classification algorithm and learn the parameters of it. This phase can be called as training. Following the training phase, the test fold (set) is used for validating the training on an unseen data (testing phase). Hence, we can validate the automatically optimized settings of a classification algorithm (training phase) with a completely new dataset. K-fold CV applies this training and testing procedure for K times. Likewise, Stratified K-fold CV while applying the K-fold Cross Validation sets, the class distribution in each fold is approximately the same as in the initial dataset. Hence, it is an appropriate method to for both accuracy estimation and model selection for a classification problem (Kohavi et al., 1995).

However, Stratified K-fold CV can give biased results as the hyper-parameter tuning is done on the same dataset. In their research Varma and Simon (2006) provide a detailed comparison of the bias in estimators, concluding that the least bias can be obtained when you separate the hyper-parameter tuning and validation steps in different loops with a nested cross-validation. Thus, while reporting the classification accuracy, nested 10-fold cross-validation with a 5-fold cross-validation for the hyper-parameter tuning will be used to minimize the bias. This will include two different processes. First with an inner cross-validation, we will tune the hyper-parameters of each classification model. Then with the optimized parameters we will run an outer cross-validation to have an aggregated error estimation for the models selected. All the experiments will include a fixed seed setting, to control the randomness. The Figure 4.2 below shows the distinction between outer and inner loops in the process of nested cross validation that we used for the validation of our model. The figure is drawn for 5-fold CV for simplification. However, in our experiments we will use 10-folds for the outer loop, instead of 5-folds. And the 5-fold CV remains the same for the inner loop.

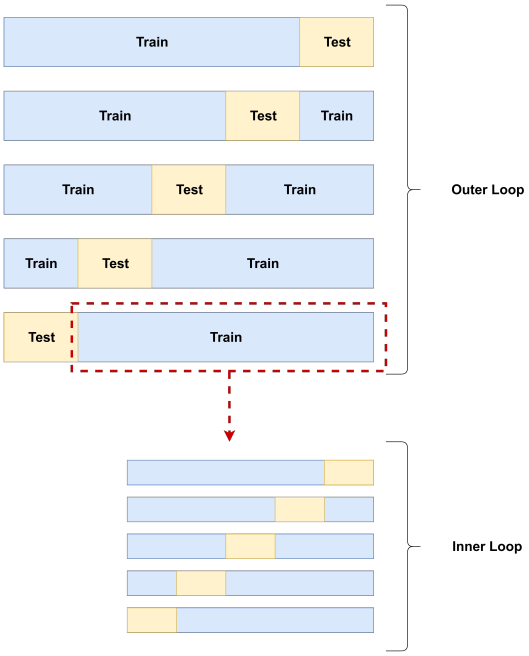


FIGURE 4.2: The nested CV logic with Outer and Inner loops.

The experiments will be run by using a fixed set of parameters. For the text and each of the classification algorithms selected we will use different settings. The settings we use in the experiments are shown in Table 4.1. For the text settings, we will use a fixed N-Grams setting that is 1 and 2-grams. Additionally, we restricted the number of the words by taking only the most frequent 4,000 n-grams for each of the experiments. We will experiment with each of the dimension reduction algorithms for text including the no DR option. The following section will provide more details on the tuning of the given fixed set of the parameters in Table 4.1 and in Table 4.2.

4.4.2 Hyper-parameter Tuning

No-free lunch (NFL) theorems of optimization imply none of the classification algorithms performs better than another one over all possible classification settings (Wolpert & Macready, 1997). As a result we cannot be sure about a superiority of a classification algorithm and its setting over the rest of the possible classification settings. To train the algorithms with optimum hyper-parameter settings we can train and test with each of the parameters and compare with each other inside of the training phase as an inner loop of CV, as explained in the previous Subsection. This way of one-by-one comparison of the parameter performance with a cross-validation procedure is called Grid Search CV. It is known as a traditional method for hyper-parameter tuning (Hsu, Chang, Lin, et al., 2003)[2].

Setting	LigthGBM	Logistic Regression	SVM
Learning Rate	[0.01, 0.1, 1]	-	-
N-Estimators	[75, 150, 300, 450]	-	-
C	-	[1, 10, 50, 100, 200, 300]	[0.001, 0.01, 0.1, 1, 10, 100, 1000]
Kernel	-	-	Linear
Penalty	-	[L_1, L_2]	-

TABLE 4.2: Classification algorithm settings for the experiments.

Table 4.2 summarizes the parameters to be tuned in the inner loop procedure. We use 3 different classification methods. While tuning the parameters we are using Grid Search to find the best performing number of dimensions for each of the classification algorithms. The number of dimensions 20, 40, 60, 80 and 100 are tested as a part of Option DR. Besides the number of dimensions for text, we optimize the most basic parameters for each classification algorithm.

- **Light GBM:** For Light GBM we are optimizing the learning rate and the number of estimators. Learning rate is a shrinkage parameter for algorithms optimization. Smaller values of learning rate is empirically known as improving the Accuracy (Friedman, 2002). However, in our initial experiments we observed an inconsistency between the optimum values of learning rate. As a result we chose 0.01, 0.1 and 1 in our experimental settings. Number of estimators in

[2]There are two other popular hyper-parameter tuning methods as Random Search (RS) and Bayesian Search (BS) (Snoek, Larochelle, & Adams, 2012; Bergstra & Bengio, 2012). Both of the methods are suggested for improving the efficiency in terms of computational time. RS randomly selects a set of hyper-parameters to compare according to a quality measurement, whereas BS improves the selection process with Gaussian Process. On the other hand, Grid Search CV provides a manual sequential search option, which improves the performance of classification. As we do have a limited set of observations and no efficiency concerns, not to decrease the classification performance, in this thesis we will be applying only Grid Search.

Light GBM indicates the number of iterations in the boosting process. This is a parameter that prevents overfitting by iterating less or more in number. We choose the minimum number of iterations as 75 and increase the number of iterations by 150 to provide a larger range for this parameter.

- **Logistic Regression:** For the Logistic Regression (LR) we optimize the parameter C and the Regularization methods (L_1 and L_2). C is defined as $1/\lambda$ explained in the Section 4.3. Larger values of C implies less regularization. In the initial experiments with hyper-parameter tuning, for the C parameter in LR, mostly the values larger than or equal to 1 were selected. Therefore we choose to optimize values between 1 and 300 for parameter C.

- **Support Vector Machines (SVM):** For the Support Vector Machines (SVM), we are applying only linear kernel and 6 different C values as a power of 10. Larger values of C indicates less regularization. In the initial experiments depending on the dataset split, optimal C parameter was observed to be inconsistent. Therefore we choose a large range (between 10^{-3} and 10^{3}) to optimize.

We use Grid Search CV with 5-folds to find the best performing setting for each of the classification algorithms that is selected to be compared in this thesis. The performance for hyper-parameter is defined with Accuracy for classification. As we use a balanced set in our experiments, we do not expect a considerable difference between Accuracy and F1-score outcomes. However, Grid Search will define the optimum hyper-parameters by only checking Accuracy results.

4.4.3 Quality Measurements

While estimating the error rates for the classification of young adult criminal cases, we will use F1-score, Area Under the Receiver Operator Characteristic Curve (AUC-ROC) and the Accuracy measurements.

F1-scores, changing between 0 and 1, provide an overview how the harmonic mean of Recall and Precision are distributed over the selected options. In the classification literature, Precision indicates the ratio of correctly assigned classes (True Positives) over the total number of assigned classes by the machine learning algorithm (True Positives and False Positives). On the other hand, Recall indicates the correct assignments made by the machine learning algorithm (True Positives) over the total number of actual positives (True Positives and False Negatives) (Yang & Liu, 1999). Hence, when we use F1-score we will get an overview of both Precision (see Equation 4.5) and Recall (see Equation 4.6). Equation 4.7 shows the F1-score calculation.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.5}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.6}$$

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.7}$$

AUC-ROC indicates that how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples (Davis & Goadrich, 2006). It is calculated by plotting True Positive Rates (Recall) and False Positive Rates (1 − Specificity[3]) outcomes from different settings of the same classification algorithm. The area under the plot defines the AUC-ROC score. Intuitively, AUC-ROC score provides the probability that the two instances are ranked correctly with respect to each other by the ranking algorithm. AUC-ROC also changes between 0 and 1 and a perfectly accurate classifier would have an AUC-ROC as 1. In the balanced binary classification problems, having an AUC-ROC score more than 50% is associated with a success rate that is better than a random guess.

Lastly, the Accuracy is defined as the number of the correctly classified both positive and negative instances over the number of whole instances in the test set. Accuracy score will provide a basic overview of how accurately classified both the Adult (A) and Juvenile Law (J) instances are. Equation 4.8 shows how Accuracy is calculated.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} \tag{4.8}$$

In summary, we use F1-score to understand whether there is an imbalance in the predictions, Accuracy to have an overall view on the prediction success and AUC-ROC score to measure the success of a possible ranking between the two classes (A and J) that we defined.

[3]Specificity is defined as True Negatives over the sum of True Negatives and False Negatives ($TN/(TN + FN)$).

Chapter 5

Research Results

In this section we analyze the results aiming to answer the research questions defined in Chapter 1. We first present the experiment results without Dimension Reduction. Then we implement the various dimension reduction methods to see the impact of those in the predictive analysis of the adolescent criminal case outcomes. As we experiment with various different settings, we first analyze the results for Path 1, 2 and 3 without Option DR. Then, we analyze the results for Path 1, 2 and 3 with Option DR. For the Option DR we apply different dimension techniques.

For all the experiments, while running the algorithms to be reproducible we set a seed of 42 to control randomness. For all the experiments Table 4.1 and Table 4.2 are taken as a reference point and Gradient Search with 5-fold CV is applied. To obtain each F1-score, AUC-ROC and Accuracy quality measurements, as explained in the Section 4.4.1 we have used 10-fold Stratified CV. By using the stored quality measurements we take the average and standard deviation to present the results.

The following sections will give a detailed analysis on the results of our experiments.

5.1 Experiments without Dimension Reduction

Experiments without dimension reduction can be partitioned into three main categories as Path 1 (Text only), Path 2 (Database features) and Path 3 (Text and database features combination), as explained in Chapter 4. While using the Path 1 we did not add the Option DR (Dimension reduction) for high dimensional text features. The goal of using such an experimental setting is to set a baseline for the dimension reduced experiments that will be discussed in the further sections.

The average test scores for Path 1 (Text only) are represented in Figure 5.1, comparing Light GBM, Logistic Regression and SVM accuracy scores for each of the possible section combination for the judicial text. The results indicate that best results for the Path 1 (Text only) are obtained for Light GBM algorithm while using only the Evidence section. The best accuracy score is 69.94% (+/- 12.0%), F1-score

is 69.28% (+/- 13.2%) and the AUC-ROC is 75.08% (+/- 12.1%). When we observe the standard deviation it is the highest in the Light GBM as well. Hence, we can conclude for Path 1 (Text only) tests without dimension reduction that although Light GBM gave the best results, it is highly varying in comparison to the tests using the Logistic Regression and SVM.

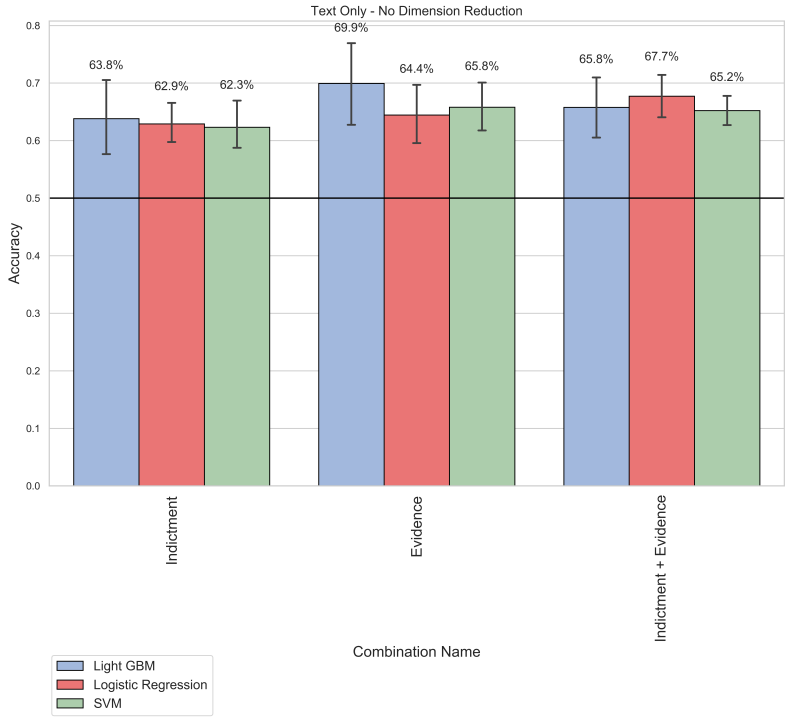


FIGURE 5.1: Comparison of the results for the Path 1 (Text only) while not reducing the dimensions of text.

Figure 5.2 shows the accuracy average test scores for the Path 2 (Database features only), comparing Light GBM, Logistic Regression and SVM for each of the possible combinations of the database feature categories (Personal and Criminal). For the tests of Path 2, the bast scores for accuracy are obtained while using Logistic Regression (67.77%). However, for the other quality measurements on average SVM performed better than all the other algorithms. The best F1-score is 69.12% (+/- 9.6%) and the best AUC-ROC is 71.89% (+/- 8.6%) in the tests with 10-fold CV.

In this Path of experiments we observed that when we combined Personal and Criminal features from the structured database we were able to have a higher performance in comparison to the experiments involving either one of the before trial database categories. Only the test results for Light GBM remained the same as the performance of the other algorithms (Logistic Regression and SVM) improved significantly with the combination of different database feature categories.

After analyzing the experiments for the Path 1 (Text only) and Path 2 (Database features only), we combined the text and the structured database features to test the

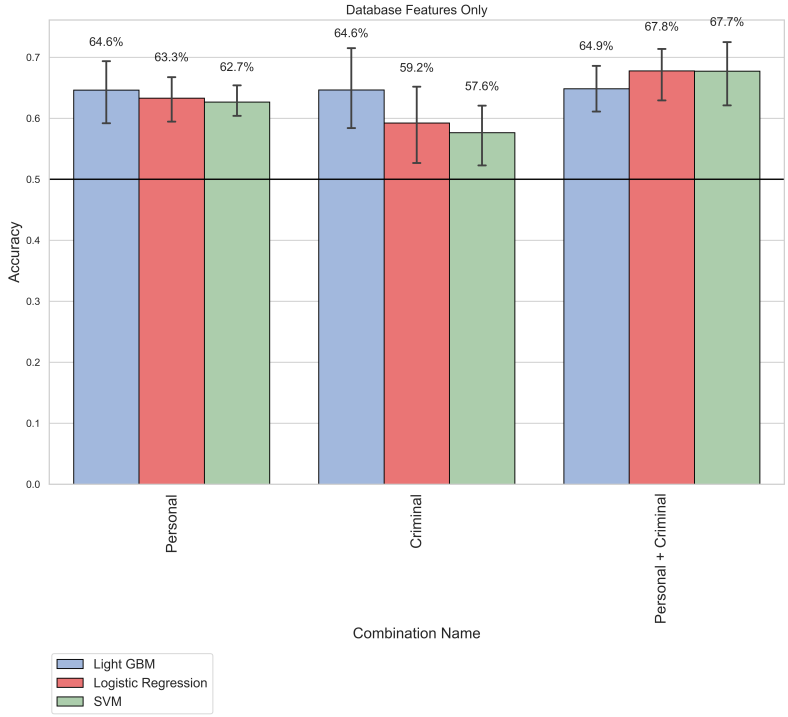


FIGURE 5.2: Comparison of the results for Path 2 (Database features only), which do not use unstructured data and dimension reduction techniques.

value added for using combination of the heterogeneous features coming from both tf-idf matrix constructed and database pre-processing.

Figure 5.3 shows the test results for each of the possible combination of database features (categories) and judicial text sections. Interestingly, the results indicate that when we combine text features and database features it improves the accuracy on average for the algorithms Logistic Regression and SVM. However, the quality measurements showed some improvement for Light GBM only while using Personal database features that are relevant for offenders' personal information and the Evidence text from the judgment text.

For the test results of Path 3 (without dimension reduction option) the best accuracy score (75.33%) is obtained when using Logistic Regression for the combination of all database features (both Personal and Criminal) and only Evidence section of the judgment text. The best AUC-ROC is 81.79% (+/- 7.5%) and the best F1-score is 74.85% (+/- 9.6%) for all the database features and Evidence section combination while using Logistic Regression.

Likewise, for SVM the best scores were obtained while using all database features and only Evidence section. For SVM the best average accuracy score is 72.25% (+/- 5.5%), the best average F1-score is 71.18% (+/- 7.8%) and the best average AUC-ROC score is 80.43% (+/- 8.0%).

The best average accuracy is 72.37% (+/- 9.2%) fot Light GBM. The best AUC-ROC is 80.74% (+/- 11.9%), which is better than the average test results from SVM. However, with Light GBM the standard deviation of the results were higher than the standard deviation of the experiments with SVM.

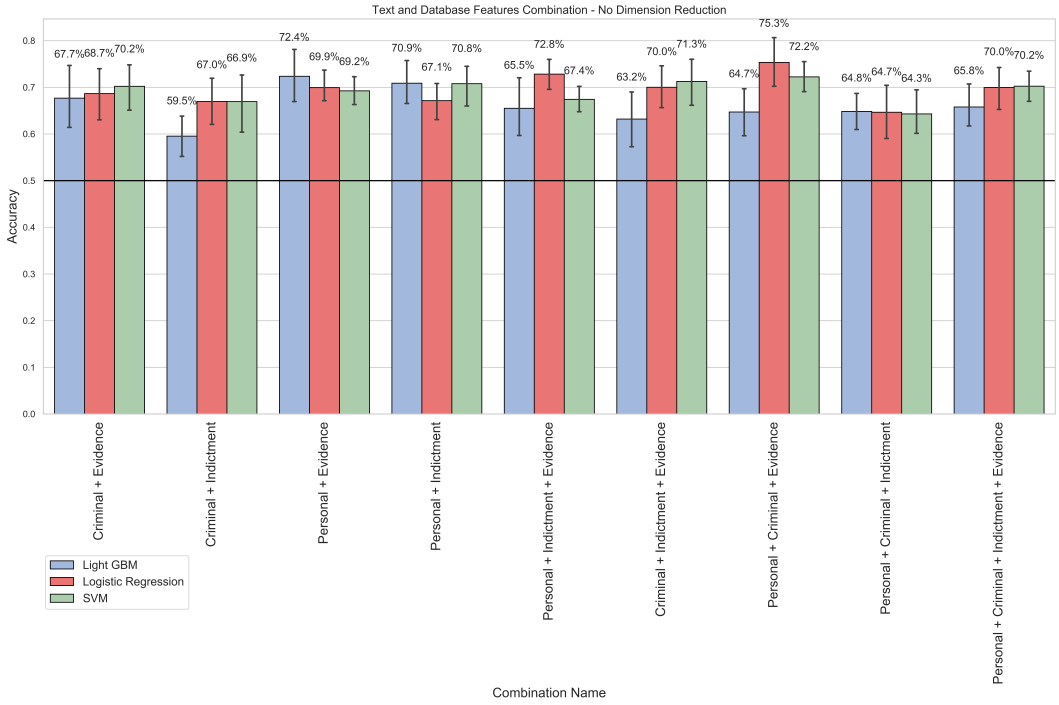


FIGURE 5.3: Comparison of the results for the Path 3 (Text and database features combination) while not reducing the dimensions of the text.

To sum up the experiments without dimension reduction gave the best accuracy results with Logistic Regression while using all the database features and only the Evidence section. Combining different data categories did not improve significantly the performance of Light GBM. Other than that we were not able to observe one leading algorithm for all the combinations.

5.2 Experiments with Dimension Reduction

In this section we will represent the test results for Path 1 (Text only) and Path 3 (Text and database features combination) as in Chapter 4, where we referred to an Option DR for dimension reduction algorithms. The goal of this series of experiments is to understand whether commonly used dimension reduction techniques for text improves the performance of the classification setting that we defined.

In this thesis we test the most common dimension reduction methods for text. In Chapter 4, where we defined those as Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) and Word2Vec. The goal of comparing multiple dimension reduction methods is to see if we can improve the classification accuracy with

matrix factorization methods, topic modeling or word embedding. In the following Subsections we will analyze the results for each of the dimension reduction methods in the given order.

5.2.1 Experiment Results with Latent Semantic Analysis

After applying nested CV with a hyper-parameter tuning for the number of dimensions and the classification algorithm parameters (explained in the Section 4.4.2), we obtained the results for both Path 1 (Text only) and Path 3 (Text and database features combination). For both of the paths we used the Option DR (dimension reduction) with Latent Semantic Analysis (LSA).

The average accuracy results for Path 1 (Text only) with Option DR (LSA) are presented in Figure 5.4. Those indicate that only for Logistic Regression while applying LSA in the section Indictment accuracy is improved in comparison to the same section combination results represented in Figure 5.1. For all the other section combinations and classifiers we observed a decrease in the average accuracy.

The best accuracy score is obtained while using Light GBM for the Path 1 (Text only) data with LSA features extracted from only the Evidence section in the judgment text. The best average accuracy score is 65.4% (+/- 10.1%). For the other metrics Light GBM is the best performing algorithm as well. The best F1-score (66.33%) is obtained while reducing the dimensions of both Indictment and Evidence sections when Light GBM is used. And the best average AUC-ROC is 72.29%.

Figure 5.5 shows the average accuracy results for the Path 3 (Text and database features combination) while using Option DR (LSA). Those report a significant increase for Path 3 when the classifier is Light GBM. For most of the combinations Light GBM became the best performing classifier when LSA was applied to the unstructured data. The best results are obtained while using all database features categorized as before trial (Personal and Criminal) and Indictment section only. The best average accuracy score is 74.77% (+/- 7.7%), the best F1-score is 74.04% (+/- 7.7%), and the best average AUC-ROC score is 80.92% (+/- 7.8%) while using the same setting. The best scores obtained in the Path 3 (Text and database features combination) while using Option DR (LSA) are slightly worse than the best scores obtained while using Path 3 without Option DR, but they have lower standard deviation.

The second best scores obtained with LSA and all possible before trial combinations are using all the possible database features (Personal and Criminal) and all the possible judgment sections (Indictment and Evidence). The second best average accuracy score while using all the possible before trial unstructured and structured data combinations is 73.5% (+/- 8.4%).

Although for Logistic Regression and SVM there is no overall improvement, with SVM while using Personal features from the database and the Evidence section the

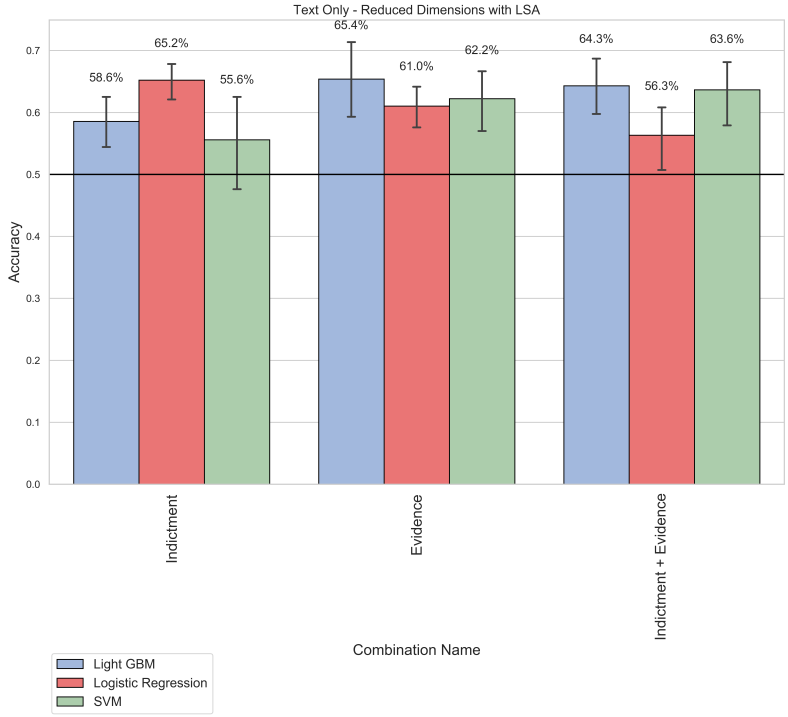


FIGURE 5.4: Comparison of the results for the Path 1 (Text only) while using LSA to reduce text features dimension.

tests report a slight improvement (an increase in the average accuracy from 69.25% to 70.58%) in comparison to the non-reduced text features test results. Furthermore in the experiments with all database features (Personal and Criminal) and their combination with Indictment section we observe a slight increase in the average accuracy scores (from 64.31% to 65.06%).

The results obtained in this section indicate that, using only unstructured features, using LSA as a dimension reduction technique does not improve the average quality measurements. However, when we combine unstructured data and structured data using the boosting algorithm (Light GBM) we can improve the average accuracy. The tests for this section were unable to demonstrate an overall improvement for the Logistic Regression and SVM.

5.2.2 Experiment Results with Latent Dirichlet Allocation

We applied 10-fold CV to obtain experiment results for Path 1 (Text only) and Path 3 (Database features and text) while using Latent Dirichlet Allocation (LDA). When we were applying the hyper-parameter tuning step (inner loop), we tune the number of dimensions that are reduced as well in the series of experiments involving LDA.

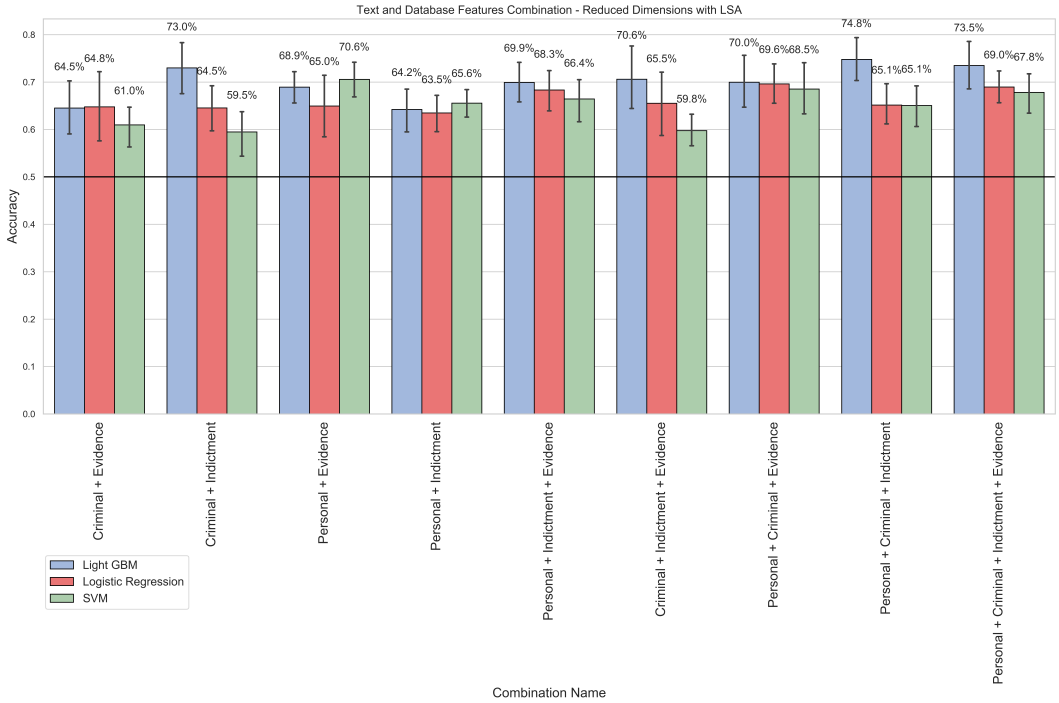


FIGURE 5.5: Comparison of the results for the Path 3 (Text and database features combination) while using LSA to reduce text features dimension.

The average accuracy results for Path 1 (Text only) with Option DR (LDA) are presented in Figure 5.6. As we observed in the Section 5.2.1, applying a dimension reduction technique does not demonstrate an improvement in the quality measurements in comparison to the results obtained while not using any dimension reduction technique. Also when we compare LDA with LSA we did not observe an average improvement when using LDA, which is extracting the topics from each of the text of judgment sections.

Figure 5.7 shows the average accuracy scores comparison between the section combinations and the classification algorithms. Although we did not see an improvement while using LDA on Path 1, while applying LDA we observed the best average accuracy measurements for all of the experiments that we have performed using Light GBM. The best average accuracy is 76.5% (+/- 7.0%) and it is obtained while combining all the database features (Personal and Criminal) and Evidence section.

Moreover, we observed an overall improvement in the quality measurements for Light GBM using Path 3 with LDA in comparison to the Light GBM using Path 3 with LSA. As we report, while using all the database features and Evidence section with Light GBM and LSA on text, we observed a 5.25% points increase in the average accuracy in comparison to the version of experiment setting not using any dimension reduction on text. Surprisingly, when we used LDA on Path 3 while using Light GBM on the all database features and Evidence we observed a 6.54% points increase in the average accuracy of the 10-fold experiments in comparison to the same setting using

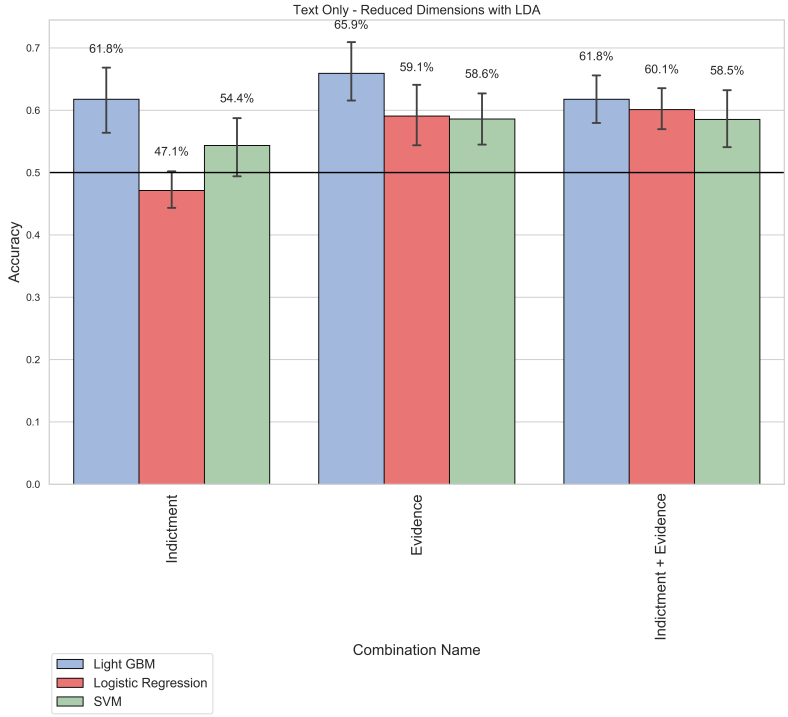


FIGURE 5.6: Comparison of the results for the Path 1 (Text only) while using LDA to reduce text features dimension.

LSA as a dimension reduction technique on text.

In summary, when we used LDA in comparison to not using dimension reduction on text, we report an average increase in the settings using both combinations of unstructured and structured data while using the boosting classifier we chose to implement. Interestingly, we also observed an average increase on the quality measurements while using LDA as a dimension reduction on text in comparison to using LSA as a dimension reduction for text. An implication might be using more topic modeling techniques might be helpful to reduce the noise in text.

5.2.3 Experiment Results with Word2Vec

We applied nested 10-fold CV for each of the classification algorithms as stated in Section 4.4.2. In the hyper-parameter step, we excluded tuning of the number of dimensions, as we were able to retrieve a strict number of dimensions (200) from the Word2Vec word embeddings. We obtained the results for both Path 1 (Text only) and Path 3 (Text and database features combination). For both of the paths we used the Option A (dimension reduction) with row-wise weighted averaged Word2Vec.

The average accuracy results for Path 1 (Text only) are displayed in Figure 5.8. The best average accuracy (66.69%) is obtained while applying Logistic Regression and Evidence section only. As we observed with LSA and LDA, applying Word2Vec

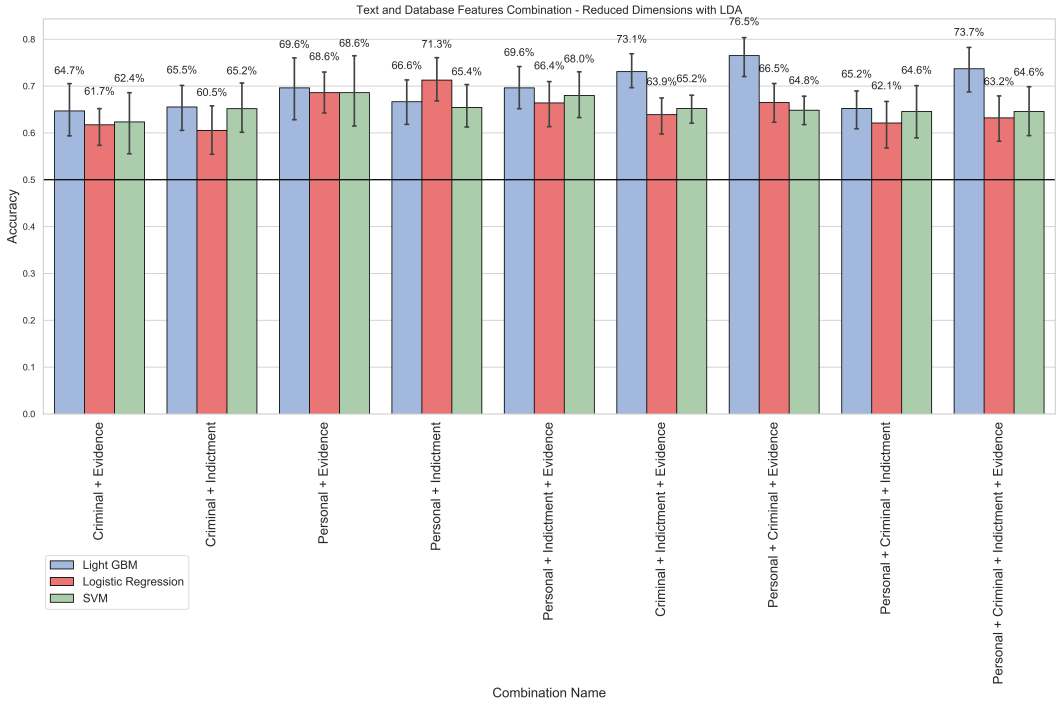


FIGURE 5.7: Comparison of the results for the Path 3 (Text and database features combination) while using LDA to reduce text features dimension.

did not improve the overall accuracy measurements in comparison to the experiments without Option DR. However, this does not mean that reducing dimensions did not add some value for the classification solution. With Word2Vec we observed the best average quality measurement scores among all the experiments using any dimension reduction technique for Path 1.

We obtained the best AUC-ROC for Path 1 (Text only) with Option DR, which is 73.25% (+/- 11.5%) while using Light GBM. Since we interpret the AUC-ROC as the probability that a randomly chosen positive example is ranked above a randomly chosen negative example, we can conclude that the best results for *ranking* the predicted instances from positive to negative is obtained while using the reduced dimensions of text with weighted averaged Word2Vec.

Figure 5.9 shows the average accuracy results for the Path 3 (Text and database features combination) while using Option DR (Word2Vec). Best scores are obtained with the before trial database features and the Evidence section combinations. The best accuracy score (73.98%) is obtained while using Word2vec with linear SVM for the combination of Personal information of the offender and the Evidence section. We did not observe an average improvement in comparison to the experiment outcomes without using Option DR.

In summary, while using Word2Vec we did not see the improvement that we were expecting from the power of word embeddings. That might be due to the amount of transformations we applied in order to use Word2Vec embeddings as a dimension

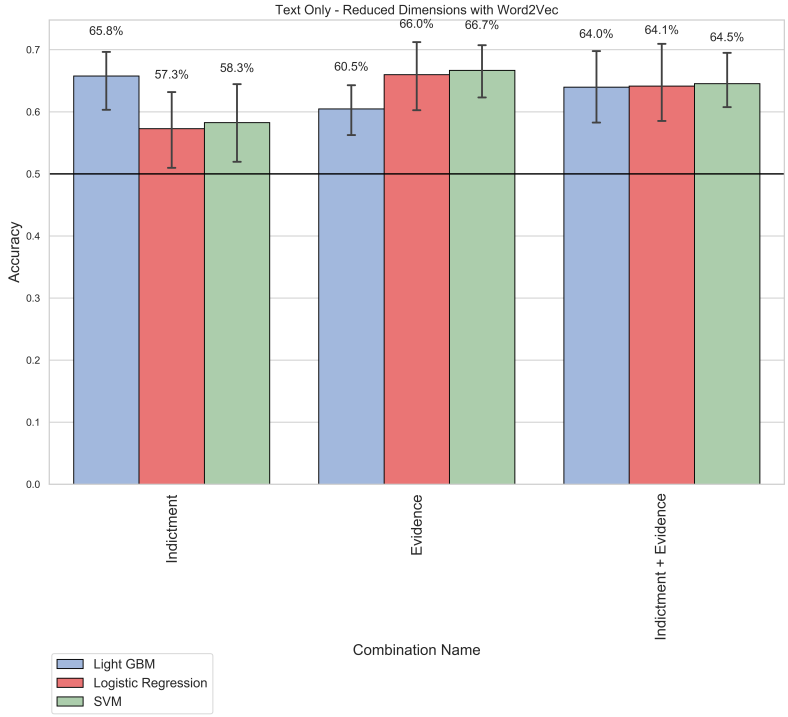


FIGURE 5.8: Comparison of the results for the Path 1 (Text only) while using Word2Vec to reduce text features dimension.

reduction method, including retrieving each word embedding and averaging row-wise using the tf-idf scores. Besides, with this method in comparison to LSA and LDA we were not able to use the power of N-grams. Thus, in case that a judgement indicates a "dangerous action" in text, we were not able to capture the meaning of combination of "dangerous" and "action" at the same time in an explicit way.

5.3 Interpretation of the Best Performing Experimental Setting

After the application of the 10-fold CV to test each of the settings (section and database feature combination) and classification algorithms we trained the pipeline (with dimension reduction technique and classifier) for the whole 316 instances and stored the pipelines. This time, instead of using a nested CV to validate the pipeline for 10 times, we tuned the hyper-parameters by using a 5-fold CV only for once and stored the model.

As we did only hyper-parameter tuning with Grid Search to find the optimum parameters for Light GBM and the number of dimensions (topics) for LDA, we were able to define the best parameters before interpreting the algorithms selected by the Grid Search. The best performing pipeline settings and its parameters were as follows:

- Light GBM used as classifier with 450 estimators and 0.01 as learning rate.

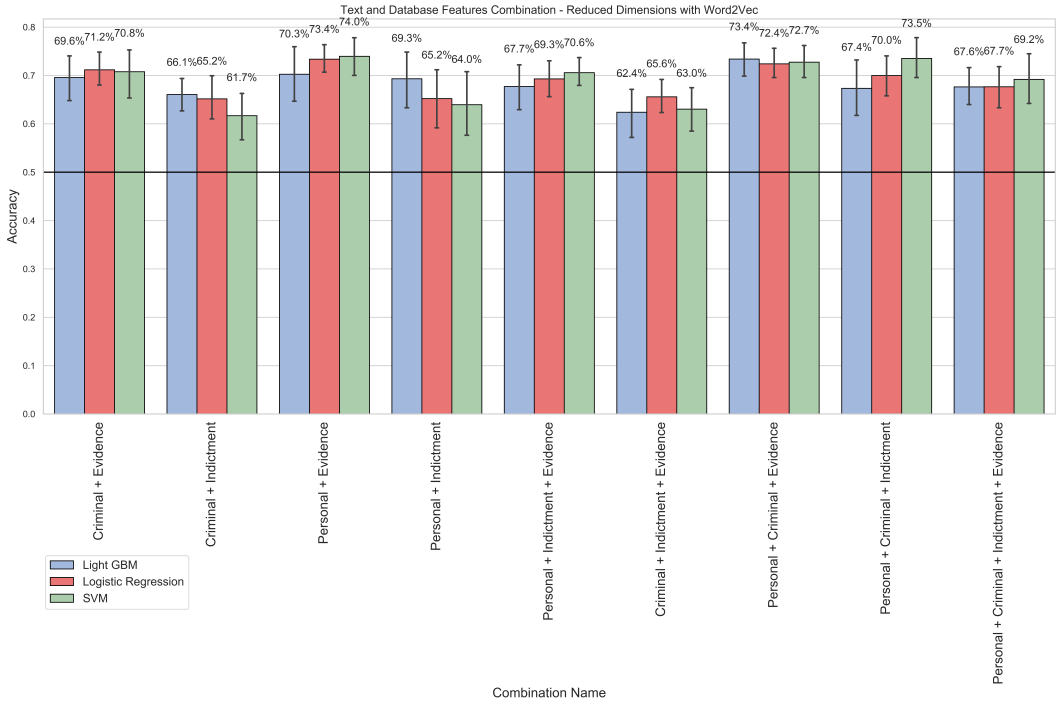


FIGURE 5.9: Comparison of the results for the Path 3 (Text and database features combination) while using Word2Vec to reduce text features dimension.

- LDA as dimension reduction technique on text with 60 topics.
- Path 3 features including all the database features and only the Evidence section of the judgments.

As we provide a complicated algorithm, we want to fill the gaps between decision making and interpretation. To fill the gap of understanding model predictions in our pipeline, we need to analyze the classification and dimension reduction methods separately as they both have different goals.

In this section we analyze the best performing pipeline setting to provide further insights for the law domain workers and public policy researchers working on the Dutch Adolescent Criminal Law. To do that we first analyze the trained boosting algorithm that is used in the pipeline and then we analyzed the most important text dimensions (topics) extracted by LDA. The following Subsections provide those analyses respectively.

5.3.1 Classification Analysis

Although the highest accuracy is obtained by a boosting mechanism, it is usually hard for domain experts to interpret boosting algorithms as those use multiple steps with weak base classifiers (another classifier) to train a predictive algorithm. The Python implementation of Light GBM and other boosting algorithms (XGBoost and

AdaBoost) provide a method for users to extract the feature importance that summarizes the features that drive the model's prediction. In those packages the feature importance for each feature is calculated by using the information gains from all the splits (for decision trees) (Hastie et al., 2017a).

In their paper Lundberg and Lee (2017b) argue that feature importance calculations are inconsistent among the features and instances. They prove that the only existing consistent way to calculate feature attributions is SHAP (SHapley Additive exPlanation) (Lundberg & Lee, 2017a).

SHAP is an additive feature attribution method that uses the expected gain calculations from the literature of cooperative game theory with the Shapley values (Lipovetsky & Conklin, 2001). One can use SHAP to interpret the feature contributions to the boosting algorithms consistently. Moreover, Lundberg and Lee (2017a) claim that the SHAP interpretations are consistent with human intuition as well. Therefore, we prefer to use SHAP as a novel algorithm to interpret our classification feature attributions.

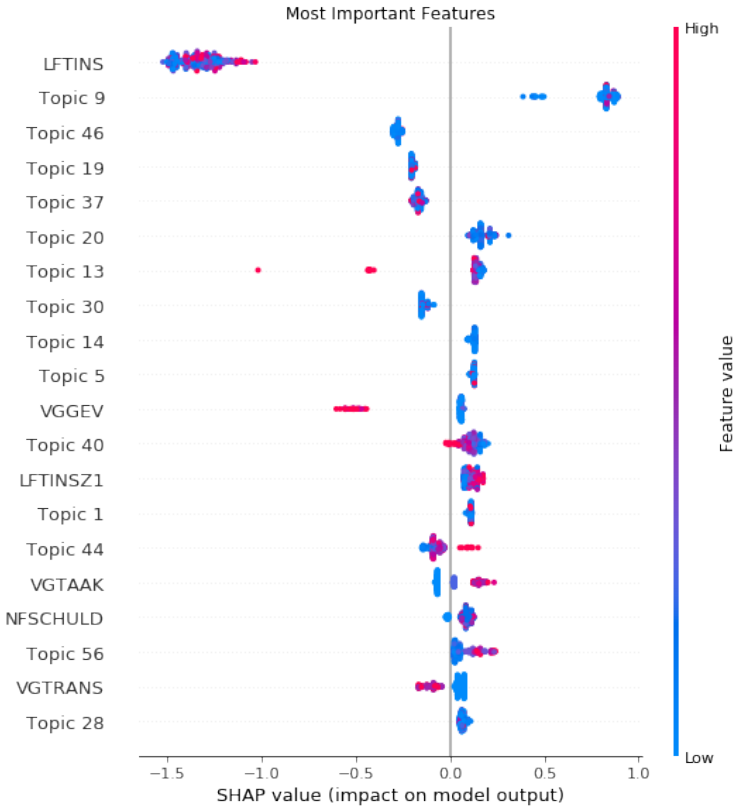


FIGURE 5.10: Most informative features in the best performing pipeline.

Figure 5.10 shows the most important features in the pipeline that we used with Light GBM. The x-axis shows the SHAP value to be interpreted. A lower value implies a higher impact on predicting a case outcome as Adult Law (negative). Therefore, for an average instance, a higher LFTINS (Age at registration in the database) increases

the probability of being judged in the Adult Law.

Another highly important feature according to their SHAP values is VGGEV, which is the number of previous custodial sentences (ordered) of the offender. As the figure suggests, if the number of previous custodial sentences are high, our model most likely predicts the outcome as Adult Law. If the number of previous custodial sentences are low, the feature has no compelling contribution.

Other database features are listed in the top 20, but others have SHAP values that are close to 0 on average. We can conclude that they have no consistent impact on the prediction outcomes. Those include LFTINSZ1 (Age at the time of first case registration), VGTAAK (Number of previous community service orders), NFSCHULD (Number of facts in the case) and VGTRANS (Number of previous transactions).

As the figure reports, in the top 20 contributing features only 6 of those are database features. Rest of the important features are reduced dimensions (topics) of Evidence section (unstructured data). We will analyze further text features (but represented as topics) in the following section. Because of the dimension reduction step, SHAP evaluates those as topics.

5.3.2 Top Contributing Topics Analysis

As we noted in the previous section most of the contributing features were LDA topics retrieved in the dimension reduction step for text data. In this section we will provide an analysis for the extracted topics and their contribution to the classification algorithm. Table 5.1 shows the most important topics appeared in the Figure 5.10 and their most relevant n-grams. All the n-grams are in the lemmatized and pre-processed form.

As we combined the insights from both the Table 5.1 and the Figure 5.10, the most important topic is Physical Damage in the Evidences. If there are less n-grams that are relevant for the Physical Damage (on an Object) topic, it is more likely to be classified as Juvenile Law. This indicates that the cases that are classified as Juvenile Law do not contain words or word combinations that are relevant to Physical Damage on an Object.

Moreover, for the Time of the Offense (Night) if there are less relevant n-grams about the night and timing, the case outcome is more likely to be predicted as Adult Law. This indicates that in case the Evidence section contains words and word combinations relevant to the Time of the Offense (Night), that case is more likely to be classified as a Juvenile Law case.

Topic Number	Topic Label	Top Relevant Keywords
9	Physical Damage (on an Object)	gevaar goed, brand ontstaan, goed duchten, vuur aanraking, stichten daarvan, vernielen beschadigen, cumulatief, inrijden, matras, onbruikbaar
46	Time of the Offense (Night)	pil, tijd woning, bestemmen tijd, nachtrust bestemmen, nachtrust, zaak adres, aangever waarbij, roos, opslaan, inbreuk
19	Criminal Organization	organisatie, waddinxveen, naam getuige, bak, nickname, nickname verdenken, zaaksdossier, samenwerkingsverband, leggen periode, crimineel organisatie
37	Language Usage (Faulty Language)	taalfout voorkomen, type taalfout, tenlasteleggging type, taalfout, schaden strafbaarheid, creditcard, kracht hoofd, dreigen woord, hand zien, hard
20	Indictment Keywords	albert heijn, albert, heijn, cumulatief alternatief, cumulatief, feit cumulatief, alternatief last, rechtbank dienen, rechthebbend bevinden, waddinxveen
13	Judgment Titles	overval, dader, tenlastegelegde, bewezenverklaarde, dienen, medeplegen, vraag, strafbaarheid feit, feit uitsluiten, meermalen
30	Explosive	tankstation, adres delict, hengelo, video, bewijsmiddel bevatten, rechtbank overtuiging, explosief, modus, ontploffing, pakket
14	Reference and Clarification	groningen, overeenkomstig bepalen, bewijsmiddel overeenkomstig, ondubbelzinnig bekend, ondubbelzinnig, duidelijk ondubbelzinnig, lid volzin, bepalen artikel, strafvordering verdenken, bepalen
5	Age of the Victim	jaar, leeftijd, zestien, zestien jaar, jaar bereiken, bereiken, twaalf, handeling plegen, echt ontuchtig, twaalf jaar
40	Case Components (Terminology)	verdenken, slachtoffer, rechtbank, medeverdachte, feit, __date__, bewijzen, verdachte, aangever, verklaren
1	Physical Damage (on a Subject)	zwaar, zwaar lichamelijk, lichamelijk letsel, lichamelijk, letsel, toebrengen, voorwaardelijk opzet, zwaar mishandeling, overtuiging, opzet
44	Evidence Collection	aangeefster, verbalisant, aantreffen, telefoon, benadelen, geven, stellen, inbraak, foto, adres
56	Procedural Information	slaan, wettelijk, opmaken, hoofd, inhoudende, d __date__, pagina, vorm opmaken
28	Punishability of the Evidences	artikel, munitie, stellen artikel, opiumwet, strafbaar stellen, strafbaarheid, wapen munitie, strafbaar, misdrijf, verenigen persoon

TABLE 5.1: LDA topics in Dutch and their labels.

The rest of the topics were less significantly and homogeneously effective on the classification model as they do not have a high absolute SHAP value. Besides, not all the high or low values were indicating the same outcome. Due to the small corpus size, most relevant word output for the LDA topics were not as indicative as we expected. However, it gives a general idea on the topic modelling features.

5.4 Discussion of the Research Results

The following discusses the findings presented in this chapter and in the Appendix D by referring to the research questions asked in Section 1.1. The discussion first answers the sub-questions, then provides a conclusion for the main research question that is: *"To what extent, we can automatically predict the adolescent criminal case outcomes by using different combinations of* ***the before trial*** *sections of the judicial texts and different types of the database information?"*

To answer the main question, we wanted to start with predicting by using only one specific type of information relevant to the adolescent criminal law cases reducing the high dimensions with a statistical technique (Option DR). We defined 3 different paths in Figure 4.1 as: Path 1 (Text only), Path 2 (Database features only) and Path 3 (Text and database features combination). For the Paths 1 and 2, we defined sections and categories to measure the contribution of each section (unstructured) or category (structured).

When we used only Path 1, we benefit more from the Evidence section, where the evidences are described by the judge, in comparison to the Indictment section, where the procedures and previous hearings are summarized by the judge. Therefore, combining both sections and retrieving the tf-idf matrix did not help to improve the quality measurements, as Indictment section was adding noisy information and we were restricting ourselves by inserting the top 4000 frequent N-grams into our matrix. The best average AUC-ROC was 76.18% (+/- 6.2%) while using only Evidence text and Linear SVM. This indicates that when we rank the predictions (as probabilities) of Juvenile and Adult Law cases, 76.18% of those which are Juvenile will be ranked above Adult Law, which is a successful result. F1-scores and Accuracy measurements were observed to be close to each other. As we used a balanced dataset, we were expecting this. Best Accuracy and F1-score were reported for Path 1 while using Evidence section and Light GBM. The best average accuracy for Path 1 was reported as 69.94% and the best average F1-score was reported as 69.28%. We can conclude that combining different sections of judgment text did not add value.

When we used only Path 2, we benefited from combining different categories of database features. Using only Personal information related to the offender resulted in better preforming predictions, in comparison to the Criminal information related to the case and the offender's criminal history. This might be a result of noisy nature

of the structured database, as Criminal database features contain many manually entered categorical information. The best average AUC-ROC was 71.89% using Linear SVM and all the available database features. Logistic Regression and Linear SVM performed similarly when we reported the accuracy results. However, Light GBM did not show an improvement over the combination of different categories. The best average Accuracy (67.77%) was reported while using Logistic Regression and all the database features, whereas the best average F1-score (69.12%) was reported while using Linear SVM and all the database features. As it is observable from the best performing experimental settings, we observed an improvement when we combined different categories of database features and used linear classification models.

We defined Path 3 to measure the impact of using unstructured and structured data while predicting the outcomes of the adolescent criminal law cases. Hence, while evaluating the results we concentrate on the combinations of structured (Personal and Criminal) and unstructured (Indictment and Evidence) data. When we used Path 3, we definitely observed an improvement on the predictions. We obtained the most accurate experiment results while combining all the before trial database features and Evidence section. The best average AUC-ROC increased from 76.18% (best average AUC-ROC in the experiments with Path 1) to 81.79% in the experiments using a linear classifier and no dimension reduction (DR) techniques for text. The best average F1-score and Accuracy were obtained from the experiments using Latent Dirichlet Allocation (LDA) as a DR technique for text and Light GBM as classifier. When we combined unstructured and structured data the best average F1-score increased from 69.28% to 77.22%. The best average Accuracy increased from 69.94% to 76.5%. Taken together, combining unstructured and structured data - with or without a dimension technique - improved our predictions, especially when we used Evidence section of judgments as unstructured data.

Lastly, we applied 3 different DR techniques to enhance the unstructured features by reducing the possible noise in it. Those techniques are defined as Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) and weighted averaged Word Embeddings with Word2Vec (Word2Vec). None of the linear classification techniques used in this thesis did benefit from a dimension reduction technique for text. We observed a decrease of quality measurements while using Logistic Regression and Linear SVM. Interestingly, Light GBM as a boosting technique benefited most from all 3 dimension techniques used in this thesis when we combined structured and unstructured data. When we used all the database features and their combinations with Evidence section (without using a DR) we were able to obtain only 70.45% as average AUC-ROC, 62.56% as average F1-score and 64.71% as average Accuracy. For the same set of features, when we applied LSA for Evidence section, average AUC-ROC increased to 78.24%. Moreover, average F1-score and Accuracy became 68.05% and 69.96% respectively. LDA improved Light GBM classifier performance - F1-score and Accuracy in particular - drastically, when we used heterogeneous data types. Average

AUC-ROC increased to 79.4%, average F1-score became 77.22% and average Accuracy, by becoming 76.5%, was observed to be the best of all. Although not as much as with LDA, we observed also an improvement in the predictions for the experiments with Word2Vec using all database features and Evidence. Average AUC-ROC was observed to be 78.61, average F1-score was reported as 72.22% and average Accuracy as 73.4%. The results for the experiments using DR confirm partially that DR improves predictions when unstructured data is used in the feature set. The most promising results were obtained while using a rather complicated classifier (Light GBM) and heterogeneous features (combination of unstructured and structured). The observed increase with de-noising the high dimensional feature set could be attributed to the overfitting nature of boosting algorithms (Mason, Baxter, Bartlett, & Frean, 2000) and the limited size of the training set we used in our experiments.

Overall, with the reported series of experiments using before trial information relevant for the adolescent criminal law cases. Table 5.2 shows a summary for the best results obtained in our experiments. When we used Light GBM and LDA as a DR for text, we were able to differentiate Juvenile Law cases from Adult Law cases with 76.5% average Accuracy. By using Logistic Regression as classifier and all the before trial database features and Evidence section as the feature set, we were able to rank 81.79% of the Juvenile Law cases before the Adult Law cases. Those results indicate a promising predictive performance in case the limitations of the feature set and this thesis is considered.

5.5 Limitations of the Study

The generalizability of the reported results is subject to certain limitations. Those are the size of the dataset, noisy features in the structured dataset, a possible sample bias in the Dutch Judiciary API, Regular Expression as a choice to distinguish the sections in the unstructured dataset, labeling strategy and class imbalance. The following will elaborate on those limitations.

One of the main concerns in our study is that we used a relatively small dataset of 316 observations. Since we use a high dimensional feature set, the small number of training size might be prone to overfitting. To improve that one can simply increase the number of observations in a similar study.

In this thesis, we have used a structured dataset retrieved from Dutch Public Prosecution Service (OM). This dataset is currently used also in the Adolescent Criminal Law research of the WODC. However, the dataset contains many redundant categories, manually entered observations and aggregated columns. This creates noise in the dataset when we use a machine learning algorithm to predict outcomes of the relevant cases. To obtain better results, Dutch Public Prosecution Service (OM) database should be better monitored and the entries should be more automatized.

Path Name	Measurement	Algorithm	Data Combination	Dimension Reduction on Text	Best Average Score
Text Only (Path 1)	Accuracy	Light GBM	Evidence	Not Applied	69.94% (12.0%)
	F1-Score	Light GBM	Evidence	Not Applied	69.28% (13.2%)
	AUC	SVM	Evidence	Not Applied	76.18% (6.2%)
DB Features Only (Path 2)	Accuracy	Logistic Regression	Personal + Criminal	Not Applied	67.77% (7.2%)
	F1-Score	SVM	Personal + Criminal	Not Applied	69.12% (9.6%)
	AUC	SVM	Personal + Criminal	Not Applied	71.89% (8.6%)
Unstructured and Structured (Path 3)	Accuracy	Light GBM	Personal + Criminal + Evidence	LDA	76.5% (7.0%)
	F1-Score	Light GBM	Personal + Criminal + Evidence	LDA	77.22% (7.0%)
	AUC	Logistic Regression	Personal + Criminal + Evidence	Not Applied	81.79% (7.5%)

TABLE 5.2: Research exepriments, best results summary.

The unstructured dataset used in this thesis was retrieved by using the Dutch Judiciary API. Dutch Judiciary API provides a *selected set* of judgments in their website (De Rechtspraak, 2014). To publish the selected judgments publicly, they provide a set of criteria (De Rechtspraak, 2012). Those criteria include following:

- published by the Supreme Court of the Netherlands
- based on an offense concerning crimes against life
- an unconditional prison sentence of four years or more and / or a provision by government (TBS) measure has been imposed
- has received public media attention, before, during or after the hearing
- the judgment has been published or discussed in a medium related to legal profession
- the judgment has a case-law character, which constitutes a precedent binding on other similar cases

With a possible sample bias in the unstructured dataset retrieved from Dutch Judiciary API, caution must be applied, as the findings in this thesis might not be transferable directly into a possible support tool for legal employee.

After retrieving the unstructured dataset we used Regular Expressions to distinguish between the different sections in judgments. Since not all the judges use a unique judgment format while defining the sections, Regular Expressions might be prone to a degree of mistakes. For instance a judge might use the word "bewezenverklaring" (in English "proven"), whereas another one might use only "bewijs" (in English "evidence"). As we retrieved only the most frequent headings, some parts of the unstructured dataset might be missing. Besides, some titles might be wrongly stated or misspelled. Therefore, Regular Expressions might not be able to differentiate the specific titles defined. To improve the precision of a future qualitative research on judgment text, judgment schema might be unified by the Dutch Judiciary. All the judges might apply the same format nation-wide.

Lastly, the dataset is labeled by the author of this thesis, who does not know Dutch. Although, a native Dutch speaker domain expert randomly sampled 30 of the cases and reported 100% accuracy for the labels, with regards to the 499 cases (without under-sampling) there might be False Positives or False Negatives due to translations. Having said that, as we used an under-sampling process to maintain a balanced dataset, the results may not be extrapolated to all criminal law cases.

To examine the label distribution of the Adolescent Criminal Law (art. 77c) cases after April 1st, 2014, we retrieved all the criminal law case judgments with Content Details and Full Text after the introduction of art. 77c till January 1st 2019. As we obtained (almost) 100% accurate classifier in the Section 6.1 with Logistic Regression, we trained it on the balanced set of 316 adolescent criminal case observations and predict the outcomes of retrieved cases. There were 16,248 cases retrieved in total, and 8.9% are predicted to have an Adolescent Criminal Law outcome. This ratio might give an indication of a highly unbalanced set. However, this must be interpreted with caution because we have observed 6 False Positives among randomly selected 20 positive (predicted) instances.

Chapter 6

Practical Results

In this chapter we analyze the results aiming to answer the practical questions defined in the Chapter 1. Therefore, we only experiment with the full datasets that are considered unstructured and structured. The unstructured dataset cover Content Details section, Full Text and their combinations. The structured dataset cover all the available categories in the database information related to the corresponding criminal case. Those categories include Personal, Criminal, Procedural and Punishments. Hence, in this chapter while running experiments we consider the outcomes, sanctions and the relevant article names as well.

In this chapter we do not apply any dimension reduction methods on text, as the non-reduced dimensions give highly accurate results. The experiments are run based on the methodology defined in the Table 4.1 and Table 4.2. To obtain each F1-score, AUC-ROC and Accuracy quality measurements, as explained in the Section 4.4.1 we have used 10-fold Stratified CV. By using the stored quality measurements we take the average and standard deviation to present the results.

The following sections will give a detailed analysis on the results of our experiments.

6.1 Experiment Results for the Practical Goal

The main practical goal of this thesis is to obtain an accuracy that is close to 100%. To do that we decide to use all the available information for 316 cases that we defined. Those cases contain equal number of Juvenile and Adult Law outcomes. Experiments for the practical goal can be partitioned into three main categories as Path 1 (Text only), Path 2 (Database features) and Path 3 (Text and database features combination), as explained in the Chapter 4.

The average accuracy scores for all the possible paths are represented in the Figure 6.1, comparing Light GBM, Logistic Regression and SVM accuracy scores for the complete available information including structured and unstructured data. The best average accuracy (99.69% +/- 1.0%) is obtained while using Content Details

section and Full Text available with Logistic Regression.

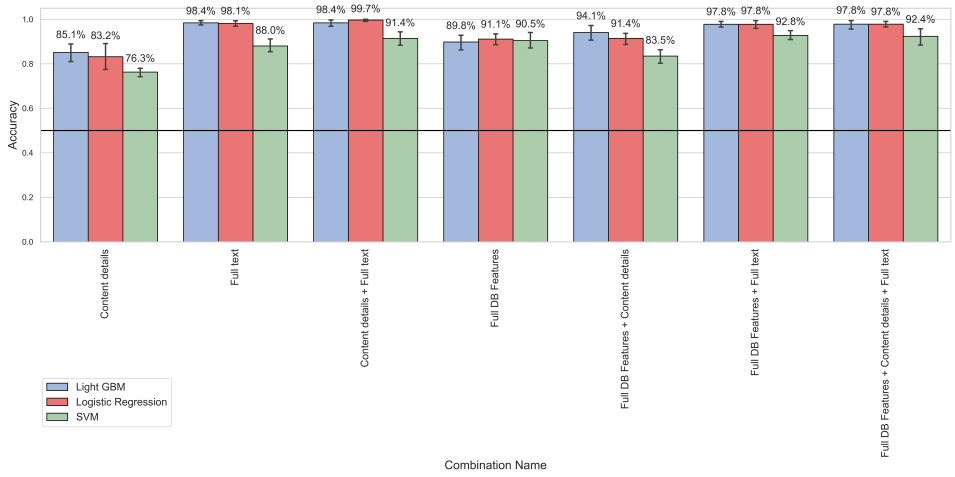


FIGURE 6.1: Comparison of the Practical Experiments.

When we compare the classifiers the worst performing one is the Linear SVM. Light GBM and Logistic Regression perform similarly successful on the given binary classification task.

Interestingly, by using only Content Details, which is a relatively short summary part of the judgments, we are able to obtain 85.08% (+/- 6.9%) of accuracy with Light GBM. On the other hand by using all the available database features we obtain 91.1% (+/- 4.3%) of accuracy. We would expect a higher accuracy, since database features include also the outcomes of the criminal law cases in a structured way. One explanation might be the high number of missing and noisy features found in the dataset.

As a consequence of the low performance while using the structured dataset, when we combine the structured data and the unstructured data we do not observe an improvement in comparison to the results while using only unstructured data. The best accuracy that we observe is 97.81% (+/- 2.1%) with Logistic Regression.

In summary when we use only all available sections of the judicial text, we obtain an average accuracy that is so close to 100%, while using Logistic Regression. As the structured dataset contains missing and noisy information, it does not add value to the binary classification accuracy.

6.2 Interpretation of the Best Performing Algorithm

We reach the practical goal of this thesis (having an accuracy close to 100%) while using Content Details and Full Text with Logistic Regression. To interpret the algorithm and the features that are used in this experiment, we train Logistic Regression on the whole 316 adolescent criminal law cases. By using that model, we

aim to see the positively and negatively effecting 1 or 2-grams in the classification of Adult or Juvenile outcomes of a given adolescent criminal case.

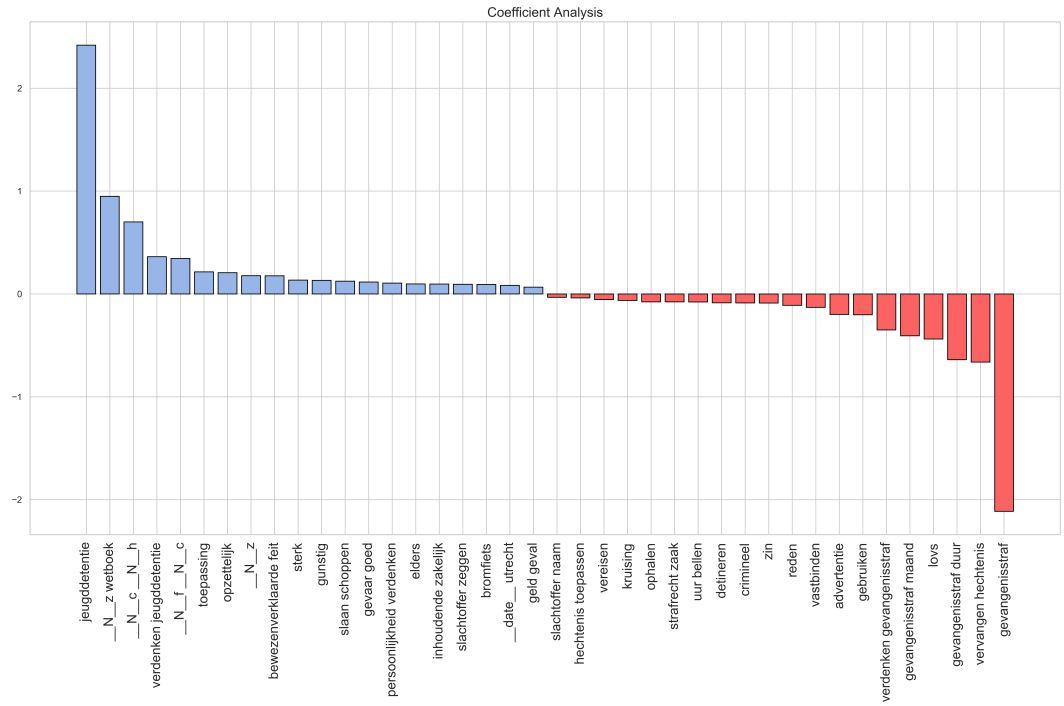


FIGURE 6.2: Coefficient analysis for the best performing algorithm.

When we say that an N-gram is positively effective, we imply that if the N-gram is used in a case judgment, that case is more likely to be classified as Juvenile Law. Conversely, when an N-gram is negatively effective, we imply that if the N-gram is used in a case judgment, that case is more likely to be classified as Adult Law.

Figure 6.2 shows the top 20 positively and negatively effective N-grams[1]. If the word *jeugddetentie* (in English *juvenile detention*) is used in either Content Details or Full Text of a criminal case, then it is 11 times more likely to be classified as Juvenile Law in comparison to a document that does not contain the word *jeugddetentie*. The other most positively related words cover some procedural words in Dutch, e.g., *toepassing* (in English *application*) and *opzettelijk* (in English *intentional*).

Turning now on the negatively effective N-grams, if the word *gevangenisstraf* is used in the judgment text of a given adolescent criminal case, then that case is 8 times more likely to be classified as Adult Law. The other most negatively effective N-grams include some sanction related Dutch words, e.g., *vervangen hechtenis* (in English *replacing custody*) and *detineren* (in English *to detain*).

[1]"__N__" in the x-axis indicates there used to be a number in text, but replaced by an empty space. Those represent the law articles related to the Adolescent and Juvenile Criminal Law.

6.3 Discussion of the Practical Results

In this chapter we wanted to answer a single technical question as follows: *To what extent, we can automatically predict the adolescent criminal case outcomes by using judicial texts and the available database information?* To understand the boundaries of this classification question we did not distinguish between before or after trial information. The main reason for this decision was to obtain an accuracy as high as possible with an interest of deploying the best performing model in the Dutch Public Prosecution Service database. The deployed algorithm was intended to be used for a better monitoring of the Adolescent Criminal Law. As long as we have accurate results, we did not trade-off between efficiency (running time) of the machine learning models and improvement of the accuracy. Therefore we did not use Option DR while answering the practical question.

The best performing predictive settings are obtained while using the whole judgment text (Content Details and Full Text combination) with Logistic Regression. 100% of AUC-ROC score in that setting shows that using Logistic Regression scores we were able to rank all the Juvenile Law cases before the Adult Law cases. However, when we applied a 50% threshold to make classification with Logistic Regression, 2 of the observations in 316 were wrongly classified (one of them is False Negative, the other one is False Positive). Therefore, we had almost the same F1-score (99.68%) and Accuracy (99.69%). As we expected expected, average accuracy results using a classification algorithm (changing between 76.27% to 99.68%) were improved in comparison to the expert system accuracy (70.5%). Since these scores are very close to 100%, we recommend the WODC to deploy this model for monitoring the application of art. 77c.

To deploy such a model in a database we need to consider the limitations stated in the Section 5.5. In addition to those concerns, we used Dutch Judiciary API to retrieve text data. As the best performing model is thought to be deployed in a structured dataset, not all the judgment text will be covered by the Dutch Judiciary API due to the selection criteria. Consequently, without a full unstructured dataset it is not possible to deploy a predictive text mining model. Future experiments and dataset improvements are therefore recommended.

As best scores for the Practical experiments are summarized in Table 6.1, one might use only the structured dataset, as it is already available, to make automated predictions in the current database of the WODC. In that case, our experiments estimate a 91.1% of Accuracy and 96.27% AUC-ROC. When we compare our results with the Expert system analysis in Section 3.3.3 (70.5% Accuracy and 58.2% F1-score), we observe a big improvement while using a machine learning method instead of an expert system. However, we think that there is room for improvement in our machine learning methodology as well. We will discuss our recommendations for improvement in Section 6.4.

Path Name	Measurement	Algorithm	Data Combination	Best Average Score
Text Only (Path 1)	Accuracy	Logistic Regression	Content Details + Full Text	100.0% (0.0%)
	F1-Score	Logistic Regression	Content Details + Full Text	99.68% (1.0%)
	AUC	Logistic Regression	Content Details + Full Text	99.69% (1.0%)
DB Features Only (Path 2)	Accuracy	Logistic Regression	Full DB Features	91.1% (4.3%)
	F1-Score	Logistic Regression	Full DB Features	91.03% (4.1%)
	AUC	Light GBM	Full DB Features	96.27% (2.8%)
Unstructured and Structured (Path 3)	Accuracy	Logistic Regression	Content Details + Full Text + Full DB Features	97.81% (2.1%)
	F1-Score	Logistic Regression	Content Details + Full Text + Full DB Features	97.82% (2.0%)
	AUC	Logistic Regression	Content Details + Full Text + Full DB Features	99.8% (0.6%)

TABLE 6.1: Practical experiments, best results summary.

6.4 Recommendations for Improvement

To have a deployable machine learning model for monitoring a specific article application as an outcome of criminal law cases, we have two main suggestions for the domain experts. These include the improvement of the availability and privacy of Dutch Judiciary API, and the reliability of the structured dataset. To start with we review the stakeholders for such an automatized algotithm to be deployed in a database.

The stakeholders are the author (programmer) of this thesis, dataset providers, Dutch government (in particular justice system employees), law professionals and accused young adults. They are the data subjects, controllers and processors with relation to this machine learning algorithm. The author and programmer of this thesis is the data processor. She retrieved the datasets from Dutch Judiciary (Rechtspraak),

The Scientific Research and Documentation Center (WODC) and Dutch Public Prosecution Service (Openbare Ministerie). Legal professionals make use the ranking as an outcome of this algorithm. Accused people in the criminal cases will be analyzed according to that ranking, as we only provide a research for a possible legal assisting tool.

As stated in Section 5.5, Dutch Judiciary API follows a set of rules to publish criminal cases publicly. Judgments provided in the API do not contain personal information of the offender and the other subjects, as they are anonymized. However, the API does not contain all the case judgments. As an improvement for further text mining research on Dutch law cases, they could provide access to a larger number of judgments.

Moreover, all the judgment text is (pseudo-)anonymized[2] by masking the natural person names. However, since we have used publicly published judgments, it is easy to search on Google and reach the names published on a news website. For instance, it is trivial to reach details of a case from our unstructured dataset, which discusses a hostage in the Dutch Broadcast Foundation (NOS) (NOS Binnenland, 2015). Consequently, we are able to identify the offender, as in the news article the offender's name was publicly released. While providing a data application with a setting that we use in this study, data processing must be complied with data protection principles (Council of European Union, 2016).

As the results show, the database data used in this thesis contain noisy and redundant information. Most of the fields are entered manually, and there is no sufficient semantics information about the scope of the columns. To improve that Dutch Public Prosecution Service can revise the semantics and collection method of each column used in the dataset. Besides, they could aggregate some of the categories together to reduce the redundancy.

In summary, by improving the quality and availability of the dataset we can transform our machine learning model to create a deployable application. However, in the current situation it is impossible to predict all the historical data with an accuracy that is close to 100% accuracy.

[2]Dutch Judiciary provides an user interface for their dataset. This dataset is told to be anonymized in their information webpage ("Anonimiseringsrichtlijnen", n.d.). Recital 26 of the General Data Protection Regulation (GDPR) defines anonymized data as *"does not relate to an identified or identifiable natural person or to personal data rendered anonymous in a manner that data subject is not or no longer perceptible"*. Therefore, as long as someone can identify a natural person by using the available information (no matter if it is directly or indirectly used) that information is not considered as anonymized. Following this logic and referring to The Anonymization Decision-Making Framework, we assume that the personal data strategy that Dutch Judiciary followed is pseudo-anonymization (Elliot, Mackey, O'Hara, & Tudor, 2016).

Chapter 7

Conclusion and Future Work

A large and growing body of literature has investigated the boundaries of predictive analysis by using before trial information to predict legal case outcomes. As in the Literature review, Section 2.1.2, Katz et al. (2017) used structured database information to predict the voting behavior of the US Supreme Court. Aletras et al. (2016) used topic modeling as a dimension reduction technique to construct features while predicting European Court of Human Rights (ECHR) case outcomes. They obtained 79% Accuracy on average, while using different articles of the ECHR (Article 3, 6 and 8). Medvedeva et al. (2018) increased the amount of articles and cases per article, with respect to Aletras et al., to reduce the statistical bias. They report a discrepancy in the Accuracy outcomes for each Article. Their results showed 75% as an average Accuracy. That might be a consequence of contextual differences between the ECHR Articles.

As our experiment results for the practical goal indicate, it is trivial to obtain a high performance by using full judgments, including punishments and decisions. To model judicial decisions with a machine learning algorithm, we cannot use a dataset, that (partly) contains the final decisions. Therefore, following the research referred above, in this thesis, we explored to what extent we can automatically predict the adolescent criminal case outcomes by using different combinations of the before trial sections of the judicial texts and different types of the database information. While predicting the outcomes we solely used Dutch Criminal Code art. 77c, whether it is applied or not.

To answer the research question we first defined the unstructured and structured datasets for before trial and their relationship to be able to merge them. Then we ran a series of experiments with different classification algorithms and dimension reduction (DR) techniques for text. With the best performing combinations of judgment sections and database features, we obtained a 76.5% Accuracy on average. To provide assistance for the law professionals, one can simply use a ranking mechanism. As an addition on the referred literature, we used AUC-ROC metric to explore the potential of ranking predicted scores. We report 81.79% as the best average AUC-ROC. Hence, we obtained similar results in comparison to the referred articles, which report their

best Accuracy as between 75% and 80%.

This thesis extends our knowledge of predictive analysis in the legal domain by introducing a predictive model (setting) with heterogeneous data combinations for criminal law cases. The results that we obtained might serve as a base for future studies.

The results of our experiments including a boosting mechanism (Light GBM) and DR techniques indicated that while combining different types of datasets, DR techniques improve the predictions for our case. This finding contributes additional evidence that suggests future research concentrating on impactful DR techniques on text while dealing with judgment text. In this thesis we only covered LSA as a matrix factorization, LDA as a topic modeling and Word2Vec as a word embedding method. A natural progression of this work would be to use different types of DR techniques on a similar study. Non-Negative Matrix Factorization as an extension of matrix factorization methods and Doc2Vec as an improvement of Word2Vec might be explored to gain further insight.

Besides of using different DR techniques for text, one might combine reduced dimensions (topics or matrix representations) with word frequencies (tf-idf matrix) as Aletras et al. (2016) do. We applied a small experiment, using LDA topics combinations with tf-idf, to provide insights for future research. We run experiments on Text only dataset (Path 1). We observed an improvement on Linear classification algorithms' performance. In our preliminary experiments we report a 69.94% of Accuracy while using Linear SVM and all the before trial text features (Indictment and Evidence).

Lastly, we used a balanced dataset while under-sampling the Adult Law cases. To improve the applicability of our findings, future research is recommended to focus more on the imbalanced classification problem of this particular criminal law application. Other researchers, who are seeking to do a similar study (combining unstructured and structured data) could use either a boosting algorithm while reducing the dimensions of text or a linear classification algorithm with full text feature matrix.

Appendix A

Tagging the Sections of Judicial Text Data

Section Name	Regex Defined
Content Details	inhoudsindicatie
Indictment	([0-9]*)?(\\.)?(de)?( )?(inhoud)? ( )?(van)?( )?(de|het)?( )?(tenlastelegging)
Evidence	([0-9]*)?(\\.)?( )?((de|het)?( ) ?(waardering|beoordeling|overwegingen| kwalificatie)?( ) ?([a-z]*)?( )?(van)?( ) ?(het|de)?( )?(bewijs|bewezen))|([0-9]*) ?(\\.)?( )?(de|het)?( )?(kwalificatie)?( )?(van)?( )?(het|de)?( ) ?((bewezenverklaring)|(bewijsoverwegingen)|(bewezenverklaarde))
Punishment	([0-9]*)?(\\.)?(de)?( )?(strafbaarheid)( )?(van)?( ) ?(de|het) ?( )?(verdachte|feit|bewezenverklaarde)| ([0-9]*)?(\\.)?((motivering|oplegging) ( )?(van)?( )?(de)?( )?(straf(fen)?)|(strafmotivering))
Decision	(([0-9]*)?(\\.)?(de)?( )?(beslissing(en)?)) |(([0-9]*)?(\\.)?((de)?()?(uitspraak)))

TABLE A.1: Regular Expressions used for tagging the sections in judicial text data.

Appendix B

Software Packages Used in Source Code

We have used Anaconda Python 3.7.1 with the required packages listed in the Table B.1. All the experiments are run in two High Performance Virtual Machines (HPC Cloud) with Ubuntu 14.04, 8CPU and 16GB RAM. Source code is published in a public GitLab repository: https://gitlab.com/nander/adolescent-cases-thesis

Package Name	Version
lxml	4.2.5
lightgbm	2.2.2
requests	2.19.1
matplotlib	3.0.1
pandas	0.23.4
numpy	1.15.1
selenium	3.14.0
xmltodict	0.11.0
nltk	3.3
seaborn	0.9.0
scipy	1.1.0
scikit_learn	0.20.3

TABLE B.1: Python packages used in the source code.

Appendix C

Section Examples with Dutch Translations

	Juvenile Law	Adult Law
Content Details	Verdachte heeft geprobeerd een prostituee onder bedreiging van een mes geld afhandig te maken. Bewezenverklaring van poging tot afpersing. De rechtbank legt op een jeugddetentie van 179 dagen met aftrek voorarrest en de PIJ-maatregel voor de duur van drie jaren voorwaardelijk met een proeftijd van 2 jaren en onder meer de voorwaarden van een klinische behandeling en begeleiding van de reclassering.	Een gevangenisstraf van 6 maanden met aftrek van het voorarrest voor een woninginbraak in vereniging gepleegd, gevolgd van geweld tegen verbalisanten. De verdachte moet daarnaast de immateriële schade van respectievelijk €200,– en €300,– vergoeden aan de benadeelde verbalisanten.

Continued from previous page

	Juvenile Law	Adult Law
Indictment	Aan verdachte is ten laste gelegd dat: hij op of omstreeks 15 januari 2016 te Eindhoven ter uitvoering van het door verdachte voorgenomen misdrijf om met het oogmerk van wederrechtelijke toe-eigening weg te nemen geld, geheel of ten dele toebehorende aan [slachtoffer] , in elk geval aan een ander of anderen dan aan verdachte, en daarbij die voorgenomen diefstal te doen voorafgaan, te doen vergezellen en/of te doen volgen van geweld...	De zaak is aanhangig gemaakt bij dagvaarding 13 februari 2017. Aan verdachte is ten laste gelegd dat hij: op of omstreeks 01 december 2016 te Oss, althans in Nederland, tezamen en in vereniging met een of meer anderen, althans alleen, met het oogmerk van wederrechtelijke toe-eigening heeft weggenomen een sieradendoos en/of sieraden en/of (buitenlands) geld, in elk geval enig goed, geheel of ten dele toebehorende aan [slachtoffer 1] , in elk geval aan een ander of anderen dan aan verdachte en/of zijn mededaders...
Evidence	De rechtbank acht, op grond van de feiten en omstandigheden die zijn vervat in de bewijsmiddelen, wettig en overtuigend bewezen, dat verdachte: (primair) op 15 januari 2016 te Eindhoven ter uitvoering van het door verdachte voorgenomen misdrijf om met het oogmerk om zich wederrechtelijk te bevoordelen door bedreiging met geweld [slachtoffer] te dwingen tot de afgifte van geld toebehorende aan [slachtoffer] , voornoemde [slachtoffer] een mes heeft voorgehouden en daarbij heeft gezegd "geef me dat geld", althans heeft geroepen "geld", terwijl de uitvoering van dat voorgenomen misdrijf niet is voltooid...	Over het ten laste gelegde geweld heeft hij zich op het standpunt gesteld dat er in de onderhavige zaak een te ver verwijderd verband bestaat tussen het gebruik van geweld door verdachte en de diefstal. Volgens de raadsman liggen de diefstal en het geweld ruimtelijk en temporeel uit elkaar. Hij wijst hierbij op de bewoordingen "gevolgd door" en "bij betrapping op heterdaad" in artikel 312, eerste lid, Sr. Aangezien er tussen de diefstal in de woning en de uiteindelijke aanhouding van verdachte 1,5 tot 2 uur heeft gezeten en de aanhouding op een andere locatie heeft plaatsgevonden..

Continued from previous page

	Juvenile Law	Adult Law
Punishment Considerations	Verdachte heeft geprobeerd het slachtoffer onder bedreiging van een mes geld afhandig te maken. Het slachtoffer bevond zich ten tijde van de beroving in een uiterst kwetsbare positie. Zij was als raamprostituee werkzaam en verdachte bezocht haar als klant. Enkel door het daadkrachtig optreden van het slachtoffer is de beroving niet voltooid. Verdachte heeft door zijn handelen een grote inbreuk gemaakt op de persoonlijke levenssfeer van het slachtoffer en haar lichamelijke integriteit aangetast...	Een inbraak in de woning veroorzaakt gevoelens van angst en onveiligheid bij de bewoners in het bijzonder en in de samenleving in het algemeen. Daarnaast brengt een woninginbraak voor de benadeelden materiële schade en overlast met zich mee. Verdachte heeft zich van dit alles niets aangetrokken. Hij heeft zich enkel laten leiden door financiële motieven. Verdachte en zijn mededaders zijn planmatig te werk gegaan en zijn met het enkele doel om een woninginbraak te plegen afgereisd naar Oss. Daarnaast heeft verdachte geweld gebruikt tegen verbalisanten tijdens zijn vlucht. Opsporingsambtenaren verdienen respect en eerbied van een ieder. De rechtbank rekent dit alles de verdachte aan...
Decision	Jeugddetentie voor de duur van 179 dagen met aftrek overeenkomstig artikel 27. Plaatsing in een inrichting voor jeugdigen voor de duur van 3 jaar voorwaardelijk met een proeftijd van 2 jaren.	Gevangenisstraf voor de duur van 6 maanden met aftrek overeenkomstig artikel 27 Wetboek van Strafrecht. Maatregel van schadevergoeding van EUR 200,00 subsidiair 4 dagen hechtenis. Legt derhalve aan verdachte op de verplichting tot betaling aan de Staat ten behoeve van het slachtoffer [verbalisant 2] van een bedrag van EUR 200,00 (zegge: tweehonderd euro), bij gebreke van betaling en verhaal te vervangen door 4 dagen hechtenis. Het bedrag bestaat uit een bedrag van EUR 200,00 immateriële schadevergoeding...

Appendix D

Experiment Results in Tables

D.1 Experiments without Dimension Reduction Techniques for Text

Results without Dimension Reduction	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Personal	71.66% (8.0%)	63.03% (9.4%)	64.62% (8.5%)	70.07% (6.4%)	64.62% (6.8%)	63.31% (6.1%)	59.96% (16.8%)	63.64% (5.0%)	62.67% (4.2%)
Criminal	71.84% (11.4%)	63.95% (11.4%)	64.65% (11.3%)	62.09% (10.0%)	59.47% (12.7%)	59.23% (10.8%)	60.56% (5.2%)	59.9% (7.3%)	57.65% (7.9%)

Continued from previous page

Results without Dimension Reduction	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Personal + Criminal	70.63% (7.5%)	63.19% (8.9%)	64.85% (6.7%)	69.83% (8.8%)	68.69% (7.0%)	**67.77% (7.2%)**	**71.89% (8.6%)**	**69.12% (9.6%)**	67.73% (8.9%)
Indictment	66.56% (10.7%)	64.99% (10.4%)	63.81% (11.3%)	69.47% (5.0%)	62.83% (6.7%)	62.92% (5.7%)	69.04% (7.7%)	62.83% (7.1%)	62.31% (7.4%)
Evidence	75.08% (12.1%)	**69.28% (13.2%)**	**69.94% (12.0%)**	71.49% (7.5%)	61.4% (11.5%)	64.44% (8.7%)	**76.18% (6.2%)**	64.44% (10.0%)	65.81% (6.9%)
Indictment + Evidence	72.64% (9.0%)	65.17% (10.5%)	65.77% (9.4%)	73.79% (3.9%)	67.09% (6.1%)	67.73% (6.4%)	74.65% (4.4%)	64.1% (4.0%)	65.21% (4.3%)
Criminal + Evidence	76.1% (12.6%)	66.57% (12.7%)	67.69% (11.4%)	73.01% (8.2%)	67.51% (12.0%)	68.67% (9.8%)	75.7% (6.8%)	70.4% (9.5%)	70.21% (8.5%)
Criminal + Indictment	64.6% (13.3%)	60.91% (7.1%)	59.54% (7.6%)	75.11% (10.3%)	65.56% (7.9%)	66.96% (8.3%)	70.79% (11.7%)	65.63% (9.0%)	66.94% (11.2%)
Personal + Evidence	80.74% (11.9%)	72.02% (9.1%)	72.37% (9.2%)	76.64% (7.7%)	68.46% (8.3%)	69.94% (5.7%)	73.53% (7.1%)	67.3% (7.0%)	69.25% (5.0%)

Continued from previous page

Results without Dimension Reduction	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Personal + Indictment	78.99% (6.3%)	70.32% (7.8%)	70.9% (7.6%)	73.43% (6.0%)	65.45% (8.0%)	67.15% (6.8%)	73.72% (7.4%)	69.68% (9.4%)	70.77% (7.6%)
Personal + Indictment + Evidence	70.28% (8.9%)	65.41% (11.7%)	65.5% (11.0%)	79.78% (6.2%)	72.93% (6.9%)	72.83% (5.3%)	76.48% (4.6%)	65.89% (5.7%)	67.44% (4.7%)
Criminal + Indictment + Evidence	68.56% (12.3%)	63.25% (12.3%)	63.21% (10.5%)	77.73% (7.3%)	70.29% (7.8%)	70.02% (7.7%)	79.29% (7.1%)	71.88% (7.7%)	71.27% (8.5%)
Personal + Criminal + Evidence	70.45% (10.4%)	62.56% (10.0%)	64.71% (8.8%)	**81.79% (7.5%)**	**74.85% (9.6%)**	**75.33% (9.3%)**	80.43% (8.0%)	71.18% (7.8%)	72.25% (5.5%)
Personal + Criminal + Indictment	72.24% (8.9%)	62.88% (9.0%)	64.83% (6.8%)	66.72% (10.5%)	64.3% (10.8%)	64.67% (9.8%)	66.26% (7.9%)	61.66% (10.4%)	64.31% (7.7%)
Personal + Criminal + Indictment + Evidence	71.34% (9.9%)	66.18% (9.1%)	65.79% (8.0%)	77.16% (5.5%)	69.19% (8.2%)	69.96% (7.8%)	78.37% (4.7%)	69.02% (8.2%)	70.25% (5.7%)

D.2 Experiments with Reduced Dimensions for Text (LSA)

Results for Reduced Dimensions (LSA)	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Indictment	65.1% (10.8%)	58.82% (7.5%)	58.56% (6.9%)	67.98% (6.1%)	62.86% (5.5%)	65.21% (4.7%)	55.21% (9.5%)	51.6% (12.8%)	55.58% (12.2%)
Evidence	72.12% (9.2%)	64.39% (10.4%)	65.4% (10.1%)	65.36% (7.6%)	59.86% (5.3%)	61.02% (5.6%)	66.33% (10.1%)	61.34% (10.8%)	62.23% (8.4%)
Indictment + Evidence	**72.29% (6.5%)**	**66.33% (8.3%)**	**64.31% (7.9%)**	64.25% (9.1%)	54.05% (10.5%)	56.33% (8.6%)	65.79% (7.8%)	63.27% (12.5%)	63.65% (9.2%)
Criminal + Evidence	69.62% (9.6%)	65.19% (10.9%)	64.52% (9.9%)	66.37% (15.5%)	62.11% (18.3%)	64.77% (12.9%)	64.06% (8.4%)	65.28% (7.5%)	60.96% (6.9%)
Criminal + Indictment	80.16% (7.8%)	73.63% (7.5%)	73.0% (9.8%)	66.09% (6.4%)	63.33% (9.4%)	64.54% (8.1%)	61.94% (8.3%)	63.14% (7.6%)	59.48% (8.0%)
Personal + Evidence	80.71% (8.3%)	67.97% (8.6%)	68.94% (5.7%)	72.45% (11.8%)	64.45% (12.7%)	64.96% (11.3%)	77.56% (5.3%)	71.69% (5.4%)	70.58% (6.2%)

Continued from previous page

Results for Reduced Dimensions (LSA)	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Personal + Indictment	70.13% (10.1%)	60.83% (11.5%)	64.23% (8.1%)	66.73% (8.7%)	63.36% (7.1%)	63.5% (6.5%)	63.41% (13.6%)	69.98% (4.0%)	65.56% (5.0%)
Personal + Indictment + Evidence	76.8% (7.6%)	70.73% (6.8%)	69.92% (7.0%)	72.68% (7.9%)	68.03% (9.0%)	68.33% (7.6%)	67.15% (15.5%)	68.93% (7.1%)	66.44% (7.6%)
Criminal + Indictment + Evidence	76.55% (8.4%)	70.75% (11.2%)	70.6% (11.1%)	66.94% (13.2%)	64.49% (12.6%)	65.52% (11.3%)	58.74% (7.2%)	59.98% (7.1%)	59.77% (5.8%)
Personal + Criminal + Evidence	**78.24% (7.0%)**	**68.05% (9.9%)**	**69.96% (9.2%)**	74.09% (7.7%)	70.0% (6.5%)	69.6% (7.0%)	75.94% (11.3%)	69.16% (8.5%)	68.52% (9.1%)
Personal + Criminal + Indictment	80.92% (7.8%)	74.04% (7.7%)	74.77% (7.7%)	68.75% (10.6%)	62.76% (11.1%)	65.15% (7.3%)	72.47% (9.2%)	67.73% (7.7%)	65.06% (7.9%)
Personal + Criminal + Indictment + Evidence	79.87% (10.8%)	73.23% (7.6%)	73.5% (8.4%)	72.75% (6.2%)	67.81% (7.6%)	68.96% (6.1%)	68.79% (7.0%)	68.86% (8.1%)	67.81% (7.4%)

D.3 Experiments with Reduced Dimensions for Text (LDA)

Results for Reduced Dimensions (LDA)	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Indictment	62.59% (13.4%)	58.36% (12.0%)	61.77% (8.9%)	48.41% (6.1%)	50.82% (6.4%)	47.15% (5.2%)	55.43% (10.3%)	47.83% (9.2%)	54.35% (7.8%)
Evidence	**67.65% (11.2%)**	**67.02% (8.2%)**	**65.94% (8.3%)**	63.4% (11.4%)	57.86% (10.8%)	59.08% (8.4%)	61.51% (8.5%)	55.12% (12.7%)	58.6% (7.2%)
Indictment + Evidence	64.73% (7.8%)	60.79% (8.4%)	61.77% (6.5%)	64.24% (7.2%)	59.47% (5.6%)	60.1% (5.6%)	58.76% (11.5%)	56.73% (11.2%)	58.54% (8.4%)
Criminal + Evidence	71.77% (11.6%)	66.51% (10.1%)	64.69% (9.7%)	61.43% (8.5%)	61.04% (9.1%)	61.73% (6.8%)	61.97% (11.3%)	56.59% (15.6%)	62.35% (11.1%)
Criminal + Indictment	73.89% (10.0%)	64.46% (9.9%)	65.52% (8.3%)	60.38% (8.9%)	58.96% (10.1%)	60.52% (8.7%)	64.67% (10.2%)	64.87% (8.9%)	65.17% (8.7%)
Personal + Evidence	73.88% (8.7%)	71.03% (10.5%)	69.6% (11.1%)	74.87% (9.6%)	68.74% (9.3%)	68.6% (7.9%)	74.6% (13.8%)	69.84% (13.2%)	68.58% (13.6%)

Continued from previous page

Results for Reduced Dimensions (LDA)	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Personal + Indictment	71.0% (7.5%)	67.33% (8.6%)	66.65% (8.0%)	73.82% (9.6%)	71.76% (7.5%)	71.27% (7.8%)	73.32% (10.8%)	66.71% (8.9%)	65.44% (7.8%)
Personal + Indictment + Evidence	76.46% (8.4%)	68.21% (9.3%)	69.6% (7.6%)	74.0% (12.6%)	66.28% (8.8%)	66.4% (8.3%)	74.32% (6.4%)	68.34% (9.0%)	67.98% (8.1%)
Criminal + Indictment + Evidence	81.02% (7.2%)	72.9% (6.5%)	73.08% (6.3%)	63.45% (8.6%)	61.48% (8.9%)	63.9% (6.5%)	68.16% (5.4%)	64.23% (4.9%)	65.19% (5.3%)
Personal + Criminal + Evidence	**79.4% (7.1%)**	**77.22% (7.0%)**	**76.5% (7.0%)**	72.32% (7.0%)	65.09% (8.6%)	66.48% (7.0%)	69.91% (5.2%)	63.53% (4.5%)	64.83% (5.2%)
Personal + Criminal + Indictment	75.05% (7.4%)	63.2% (10.4%)	65.23% (7.1%)	63.99% (9.3%)	61.06% (10.5%)	62.1% (8.4%)	64.29% (12.5%)	63.29% (10.6%)	64.56% (10.1%)
Personal + Criminal + Indictment + Evidence	78.87% (8.6%)	73.72% (9.2%)	73.69% (8.1%)	67.58% (8.7%)	61.27% (9.8%)	63.21% (8.0%)	69.65% (11.7%)	64.13% (10.8%)	64.56% (9.3%)

D.4 Experiments with Reduced Dimensions for Text (Word2Vec)

Results for Reduced Dimensions (Word2Vec)	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Indictment	69.39% (11.4%)	64.63% (9.4%)	65.77% (7.2%)	56.29% (12.9%)	56.23% (11.3%)	57.29% (10.1%)	57.48% (9.9%)	57.56% (11.0%)	58.27% (10.3%)
Evidence	66.99% (9.0%)	60.72% (5.1%)	60.46% (6.3%)	67.13% (10.0%)	61.98% (11.3%)	65.98% (9.1%)	66.37% (10.8%)	63.34% (8.6%)	66.69% (7.3%)
Indictment + Evidence	**73.25% (11.5%)**	**64.9% (8.5%)**	**63.98% (9.9%)**	62.96% (10.8%)	62.9% (10.9%)	64.15% (10.8%)	62.95% (9.9%)	63.83% (8.4%)	64.54% (7.7%)
Personal + Indictment	77.39% (10.2%)	66.68% (14.0%)	69.33% (9.3%)	69.65% (12.4%)	65.19% (9.9%)	65.23% (10.6%)	70.07% (10.8%)	64.4% (10.3%)	63.96% (11.3%)
Personal + Evidence	78.26% (9.5%)	69.16% (10.4%)	70.27% (9.2%)	75.86% (6.4%)	71.55% (6.4%)	73.38% (4.9%)	78.11% (5.3%)	**73.3% (6.6%)**	**73.98% (6.8%)**
Criminal + Indictment	72.57% (7.1%)	65.51% (6.4%)	66.06% (5.9%)	66.99% (8.5%)	64.58% (8.4%)	65.17% (8.1%)	64.62% (10.6%)	61.57% (7.1%)	61.69% (8.3%)

Continued from previous page

Results for Reduced Dimensions (Word2Vec)	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Criminal + Evidence	78.45% (8.1%)	68.66% (8.2%)	69.58% (7.6%)	72.55% (7.6%)	70.67% (6.9%)	71.19% (6.0%)	71.05% (9.7%)	69.76% (9.3%)	70.79% (8.5%)
Personal + Indictment + Evidence	76.25% (9.3%)	67.98% (8.4%)	67.73% (8.1%)	72.96% (7.2%)	67.66% (8.0%)	69.31% (6.7%)	72.76% (7.8%)	69.42% (5.8%)	70.58% (5.0%)
Criminal + Indictment + Evidence	71.72% (7.8%)	60.44% (12.5%)	62.42% (8.6%)	67.08% (7.3%)	64.0% (8.7%)	65.6% (6.2%)	63.9% (9.5%)	61.16% (7.2%)	63.04% (7.4%)
Personal + Criminal + Indictment	73.81% (11.5%)	66.86% (9.1%)	67.35% (9.5%)	75.19% (7.4%)	68.19% (7.7%)	70.0% (7.1%)	75.6% (9.4%)	72.33% (7.9%)	73.52% (7.1%)
Personal + Criminal + Evidence	**78.61% (9.9%)**	72.22% (7.2%)	73.4% (5.9%)	76.3% (6.1%)	70.42% (6.3%)	72.42% (5.2%)	76.93% (4.2%)	71.26% (6.8%)	72.75% (5.7%)
Personal + Criminal + Indictment + Evidence	74.36% (8.5%)	66.67% (7.9%)	67.65% (7.1%)	71.74% (8.2%)	65.89% (8.7%)	67.67% (7.4%)	73.13% (8.3%)	67.71% (9.9%)	69.21% (9.0%)

D.5 Experiments to Answer Practical Questions

Results for Practical Questions	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Content Details	93.55% (2.6%)	85.22% (6.9%)	85.08% (6.9%)	91.27% (6.7%)	84.36% (9.1%)	83.19% (9.9%)	84.73% (3.5%)	77.46% (4.1%)	76.27% (3.3%)
Full Text	99.57% (0.9%)	98.41% (1.7%)	98.42% (1.7%)	99.84% (0.4%)	98.06% (2.2%)	98.1% (2.2%)	96.91% (1.6%)	87.82% (5.1%)	88.02% (4.9%)
Content Details + Full Text	99.73% (0.6%)	98.34% (2.8%)	98.44% (2.7%)	100.0% (0.0%)	99.68% (1.0%)	99.69% (1.0%)	97.84% (1.9%)	91.16% (5.7%)	91.4% (5.1%)
All Database Information	96.27% (2.8%)	89.89% (5.0%)	89.83% (5.6%)	95.4% (3.3%)	91.03% (4.1%)	91.1% (4.3%)	93.65% (3.7%)	89.89% (6.2%)	90.46% (5.8%)
All Database Information + Content Details	98.83% (2.2%)	94.06% (6.0%)	94.06% (6.0%)	97.86% (2.3%)	90.94% (4.9%)	91.44% (4.2%)	91.16% (2.1%)	84.01% (4.3%)	83.5% (5.0%)
All Database Information + Full Text	99.69% (0.6%)	97.8% (2.1%)	97.79% (2.1%)	99.75% (0.4%)	97.8% (2.9%)	97.79% (3.0%)	98.22% (1.3%)	92.67% (3.2%)	92.75% (3.4%)

Results for Practical Questions	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
All Database Information + Content Details+ Full Text	99.8% (0.4%)	97.72% (3.4%)	97.81% (3.3%)	99.8% (0.6%)	97.82% (2.0%)	97.81% (2.1%)	98.18% (2.2%)	92.0% (6.7%)	92.35% (6.2%)

Appendix E

Comparison of Results Using DR

E.1 Difference Between LSA and NDR Experiments

Difference Between LSA and NDR	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Indictment	1.46	6.17	5.25	1.49	**-0.03**	**-2.29**	13.83	11.23	6.73
Evidence	2.96	4.89	4.54	6.13	1.54	3.42	9.85	3.1	3.58
Indictment + Evidence	0.35	**-1.16**	1.46	9.54	13.04	11.4	8.86	0.83	1.56
Criminal + Evidence	6.48	1.38	3.17	6.64	5.4	3.9	11.64	5.12	9.25

Continued from previous page

Comparison Between LSA and NDR	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Criminal + Indictment	**-15.56**	**-12.72**	**-13.46**	9.02	2.23	2.42	8.85	2.49	7.46
Personal + Evidence	0.03	4.05	3.43	4.19	4.01	4.98	**-4.03**	**-4.39**	**-1.33**
Personal + Indictment	8.86	9.49	6.67	6.7	2.09	3.65	10.31	**-0.3**	5.21
Personal + Indictment + Evidence	**-6.52**	**-5.32**	**-4.42**	7.1	4.9	4.5	9.33	**-3.04**	1
Criminal + Indictment + Evidence	**-7.99**	**-7.5**	**-7.39**	10.79	5.8	4.5	20.55	11.9	11.5
Personal + Criminal + Evidence	**-7.79**	**-5.49**	**-5.25**	7.7	4.85	5.73	4.49	2.02	3.73
Personal + Criminal + Indictment	**-8.68**	**-11.16**	**-9.94**	**-2.03**	1.54	**-0.48**	**-6.21**	**-6.07**	**-0.75**
Personal + Criminal + Indictment + Evidence	**-8.53**	**-7.05**	**-7.71**	4.41	1.38	1	9.58	0.16	2.44

[a] With a given set of data combinations, negative values indicate an improvement while using LSA in comparison to not using a DR technique.

[b] Differences are given as percentage points.

E.2 Difference Between LDA and NDR Experiments

Difference Between LDA and NDR	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Indictment	3.97	6.63	2.04	21.06	12.01	15.77	13.61	15	7.96
Evidence	7.43	2.26	4	8.09	3.54	5.36	14.67	9.32	7.21
Indictment + Evidence	7.91	4.38	4	9.55	7.62	7.63	15.89	7.37	6.67
Criminal + Evidence	4.33	0.06	3	11.58	6.47	6.94	13.73	13.81	7.86
Criminal + Indictment	**-9.29**	**-3.55**	**-5.98**	14.73	6.6	6.44	6.12	0.76	1.77
Personal + Evidence	6.86	0.99	2.77	1.77	**-0.28**	1.34	**-1.07**	**-2.54**	0.67
Personal + Indictment	7.99	2.99	4.25	**-0.39**	**-6.31**	**-4.12**	0.4	2.97	5.33
Personal + Indictment + Evidence	**-6.18**	**-2.8**	**-4.1**	5.78	6.65	6.43	2.16	**-2.45**	**-0.54**
Criminal + Indictment + Evidence	**-12.46**	**-9.65**	**-9.87**	14.28	8.81	6.12	11.13	7.65	6.08
Personal + Criminal + Evidence	**-8.95**	**-14.66**	**-11.79**	9.47	9.76	8.85	10.52	7.65	7.42

Continued from previous page

Comparison Between LDA and NDR	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Personal + Criminal + Indictment	**-2.81**	**-0.32**	**-0.4**	2.73	3.24	2.57	1.97	**-1.63**	**-0.25**
Personal + Criminal + Indictment + Evidence	**-7.53**	**-7.54**	**-7.9**	9.58	7.92	6.75	8.72	4.89	5.69

[a] With a given set of data combinations, negative values indicate an improvement while using LDA in comparison to not using a DR technique.

[b] Differences are given as percentage points.

E.3 Difference Between Word2Vec and NDR Experiments

Difference Between Word2Vec and NDR	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Indictment	**-2.83**	0.36	**-1.96**	13.18	6.6	5.63	11.56	5.27	4.04
Evidence	8.09	8.56	9.48	4.36	**-0.58**	**-1.54**	9.81	1.1	**-0.88**
Indictment + Evidence	**-0.61**	0.27	1.79	10.83	4.19	3.58	11.7	0.27	0.67
Criminal + Evidence	**-1.29**	**-0.11**	**-1.64**	3.36	2.32	3.44	5.63	6	6.25
Criminal + Indictment	**-13.66**	**-8.25**	**-10.73**	**-0.75**	**-5.99**	**-6.42**	**-7.32**	**-7.67**	**-7.04**
Personal + Evidence	4.49	4.04	4.64	3.68	0.8	0.63	0.77	**-2.12**	**-1.33**
Personal + Indictment	6.42	4.81	4.84	6.44	0.87	1.98	9.1	8.11	9.08
Personal + Indictment + Evidence	**-8.17**	**-3.25**	**-4.08**	7.23	2.26	1.64	5.43	**-3.87**	**-3.35**
Criminal + Indictment + Evidence	**-3.16**	2.81	0.79	10.65	6.29	4.42	15.39	10.72	8.23
Personal + Criminal + Evidence	**-3.36**	**-4.3**	**-2.64**	6.6	6.66	5.33	4.83	**-1.15**	**-1.27**

Continued from previous page

Comparison Between Word2Vec and NDR	Light GBM			Logistic Regression			SVM		
Combination	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy
Personal + Criminal + Indictment	**-6.37**	**-9.34**	**-8.57**	**-9.58**	**-6.12**	**-7.75**	**-10.67**	**-9.6**	**-8.44**
Personal + Criminal + Indictment + Evidence	**-3.02**	**-0.49**	**-1.86**	5.42	3.3	2.29	5.24	1.31	1.04

[a] With a given set of data combinations, negative values indicate an improvement while using Word2Vec in comparison to not using a DR technique.

[b] Differences are given as percentage points.

Bibliography

Aggarwal, C. C. (2018a). Linear classification and regression for text. In *Machine learning for text* (pp. 159–207). Springer.

Aggarwal, C. C. (2018b). Matrix factorization and topic modeling. In *Machine learning for text* (pp. 31–73). Springer.

Aggarwal, C. C. (2018c). Text classification: Basic models. In *Machine learning for text* (pp. 113–157). Springer.

Aggarwal, C. C. & Zhai, C. X. (2012a). A survey of text classification algorithms. In *Mining text data* (pp. 163–222). Springer Publishing Company, Incorporated.

Aggarwal, C. C. & Zhai, C. X. (2012b). A survey of text summarization techniques. In *Mining text data* (pp. 43–77). Springer Publishing Company, Incorporated.

Aggarwal, C. C. & Zhai, C. X. (2012c). Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond. In *Mining text data* (pp. 129–163). Springer Publishing Company, Incorporated.

Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., & Lampos, V. (2016). Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, *2*, e93.

Ander, N. (2019). Adolescent-cases. https://gitlab.com/nander/adolescent-cases-thesis/tree/master/src/machine_learning/stopwords. GitLab.

Anneveldt-Committee. (1982). *Sanctierecht voor jeugdigen.*

Anonimiseringsrichtlijnen. (n.d.). Retrieved October 4, 2018, from https://www.rechtspraak.nl/Uitspraken-en-nieuws/Uitspraken/Paginas/Anonimiseringsrichtlijnen.aspx

Barendregt, C. S. & Laan, A. M. V. D. (2018). Neuroscientific insights and the dutch adolescent criminal law: A brief report. *Journal of Criminal Justice*.

Bergstra, J. & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(Feb), 281–305.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Bosch, A. v. d., Busser, B., Canisius, S., & Daelemans, W. (2007). An efficient memory-based morphosyntactic tagger and parser for dutch. *LOT Occasional Series*, *7*, 191–206.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... Grobler, J., et al. (2013). Api design for machine learning software: Experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*.

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

Chieze, E., Farzindar, A., & Lapalme, G. (2010). An automatic system for summarization and information extraction of legal information. In *Semantic processing of legal texts* (pp. 216–234). Springer.

Council of European Union. (2016). Council regulation (EU) no 679/2016. https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1528874672298&uri=CELEX:02016R0679-20160504.

Council of State Governments Justice Center. (2015). Reducing recidivism and improving other outcomes for young adults in the juvenile and adult criminal justice systems. The Council of State Governments Justice Center.

Davis, J. & Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on machine learning* (pp. 233–240). ACM.

Dawes, R. M. (1979). The robust beauty of improper linear models in decision making. *American psychologist*, *34*(7), 571.

De Rechtspraak. (1999). Uitspraken. Retrieved October 4, 2018, from https://uitspraken.rechtspraak.nl/

De Rechtspraak. (2012). Selectiecriteria. Retrieved April 2, 2019, from https://www.rechtspraak.nl/Uitspraken-en-nieuws/Uitspraken/Paginas/Selectiecriteria.aspx

De Rechtspraak. (2014). Open data van de rechtspraak. Retrieved January 1, 2019, from https://www.rechtspraak.nl/Uitspraken-en-nieuws/Uitspraken/Paginas/Open-Data.aspx

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391–407.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1–15). Springer.

Eirinaki, M., Vazirgiannis, M., & Varlamis, I. (2003). Sewep: Using site semantics and a taxonomy to enhance the web personalization process. In *Proceedings of the ninth acm sigkdd international conference on knowledge discovery and data mining* (pp. 99–108). ACM.

Elliot, M., Mackey, E., O'Hara, K., & Tudor, C. (2016). *The anonymisation decision-making framework*. UKAN.

Evans, M., Mcintosh, W., Lin, J., & Cates, C. (2007). Recounting the courts? applying automated content analysis to enhance empirical legal research. *Journal of Empirical Legal Studies*, *4*(4), 1007–1039.

Farrington, D. P., Loeber, R., & Howell, J. C. (2012). Young adult offenders: The need for more effective legislative options and justice processing. *Criminology and Public Policy*, *11*(4), 729–750.

Farzindar, A. & Lapalme, G. (2004). Legal text summarization by exploration of the thematic structure and argumentative roles. *Text Summarization Branches Out*.

François, T. & Miltsakaki, E. (2012). Do nlp and machine learning improve traditional readability formulas? In *Proceedings of the first workshop on predicting and improving text readability for target reader populations* (pp. 49–57). Association for Computational Linguistics.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), 367–378.

Gaustad, T. (2004). *Linguistic knowledge and word sense disambiguation* (Doctoral dissertation, University of Groningen).

Gillman, H. (2001). What's law got to do with it? judicial behavioralists test the "legal model" of judicial decision making. *Law & Social Inquiry*, *26*(2), 465–504.

Gittens, A., Achlioptas, D., & Mahoney, M. W. (2017). Skip-gram-zipf+ uniform= vector additivity. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (Vol. 1, pp. 69–76).

Glaser, D. (1974). Remedies for the key deficiency in criminal justice evaluation research. *Journal of Research in Crime and Delinquency*, *11*(2), 144–154.

Greenberg, D. F. (1977). Delinquency and the age structure of society. *Contemporary crises*, *1*(2), 189–223.

Hachey, B. & Grover, C. (2006). Extractive summarisation of legal texts. *Artificial Intelligence and Law*, *14*(4), 305–345.

Hagan, J. (1991). Destiny and drift: Subcultural preferences, status attainments, and the risks and rewards of youth. *American Sociological Review*, 567–582.

Hall, M. A. & Wright, R. F. (2008). Systematic content analysis of judicial opinions. *Cal. L. Rev. 96*, 63.

Han, J., Pei, J., & Kamber, M. (2011). Data preprocessing. In *Data mining: Concepts and techniques* (pp. 83–125). Elsevier.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2017a). Boosting and additive trees. In *The elements of statistical learning: Data mining, inference, and prediction* (pp. 337–387). Springer.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2017b). Linear methods for classification. In *The elements of statistical learning: Data mining, inference, and prediction* (pp. 101–138). Springer.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2017c). Linear methods for regression. In *The elements of statistical learning: Data mining, inference, and prediction* (pp. 43–94). Springer.

Hirschi, T. & Gottfredson, M. (1983). Age and the explanation of crime. *American journal of sociology*, *89*(3), 552–584.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the fifteenth conference on uncertainty in artificial intelligence* (pp. 289–296). Morgan Kaufmann Publishers Inc.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification.

Husbands, P., Simon, H., & Ding, C. H. (2001). On the use of the singular value decomposition for text retrieval. *Computational information retrieval*, *5*, 145–156.

Ioannides, A. (2017). Machine learning pipelines for r. Retrieved from https://alexioannides.com/2017/05/08/machine-learning-pipelines-for-r/

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). Classification. In *An introduction to statistical learning: With applications in r* (pp. 127–175). Springer Publishing Company, Incorporated.

Katz, D. M., Bommarito II, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the supreme court of the united states. *PloS one*, *12*(4), e0174698.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems* (pp. 3146–3154).

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, *2*, pp. 1137–1145). Montreal, Canada.

Kos, W., Schraagen, M., Brinkhuis, M., & Bex, F. (2017). Classification in a skewed online trade fraud complaint corpus. In *Preproceedings of the 29th benelux conference on artificial intelligence november 8–9, 2017 in groningen, the netherlands* (pp. 172–183).

Landauer, T. K. & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259–284.

Levenberg, A., Simpson, E., Roberts, S., & Gottlob, G. (2013). Economic prediction using heterogeneous data streams from the world wide web. In *Ecml workshop on scalable methods in decision making* (Vol. 2, pp. 12–15).

Liefaard, T. (2016). Juveniles in transition from juvenile justice to adult criminal justice. *Persisters and Desisters in Crime from Adolescence into Adulthood: Explanation, Prevention and Punishment*, 159.

Lipovetsky, S. & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, *17*(4), 319–330.

Lundberg, S. M. & Lee, S.-I. (2017a). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765–4774).

Lundberg, S. M. & Lee, S.-I. (2017b). Consistent feature attribution for tree ensembles. *arXiv preprint arXiv:1706.06060*.

Mandal, A., Chaki, R., Saha, S., Ghosh, K., Pal, A., & Ghosh, S. (2017). Measuring similarity among legal court case documents. In *Proceedings of the 10th annual acm india compute conference on zzz* (pp. 1–9). ACM.

Martin, A. D., Quinn, K. M., Ruger, T. W., & Kim, P. T. (2004). Competing approaches to predicting supreme court decision making. *Perspectives on Politics*, *2*(4), 761–767.

Mason, L., Baxter, J., Bartlett, P. L., & Frean, M. R. (2000). Boosting algorithms as gradient descent. In *Advances in neural information processing systems* (pp. 512–518).

McCullagh, P. & Nelder, J. A. (1989). *Generalized linear models*. CRC press.

Medvedeva, M., Vols, M., & Wieling, M. (2018). Judicial decisions of the european court of human rights: Looking into the crystal ball. *C onference on Empirical Legal Studies–Europe*.

Meritt, D. (1989). Introduction. In *Building expert systems in prolog* (pp. 1–8). Springer.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Moffitt, T. E. (1993). Adolescence-limited and life-course-persistent antisocial behavior: A developmental taxonomy. *Psychological review*, *100*(4), 674–701.

Moses, L. B. & Chan, J. (2014). Using big data for legal and law enforcement decisions: Testing the new tools. *UNSWLJ*, *37*, 643.

Muthukadan, B. (2018). Selenium with Python. Retrieved October 4, 2018, from https://selenium-python.readthedocs.io/

Narayan, N. (2019). Embeddings for dutch judiciary project. https://github.com/palindrome6/final-project-pdeng/tree/master/embeddings/word2vec. Github.

NOS Binnenland. (2015). Gijzeling NOS is nieuws van Australië tot Zwitserland. https://nos.nl/artikel/2016260-gijzeling-nos-is-nieuws-van-australie-tot-zwitserland.html.

Olsen, H. P. & Küçüksu, A. (2017). Finding hidden patterns in ecthr's case law: On how citation network analysis can improve our knowledge of ecthr's article 14 practice. *International Journal of Discrimination and the Law*, *17*(1), 4–22.

Opijnen, M. (2011). European case law identifier: Indispensable asset for legal information retrieval.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011a). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Dubourg, V., et al. (2011b). Scikit-learn: Machine learning in python. *Journal of machine learning research*, *12*(Oct), 2825–2830.

Pennington, N. & Hastie, R. (1992). Explaining the evidence: Tests of the story model for juror decision making. *Journal of personality and social psychology*, *62*(2), 189.

Rahm, E. & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull. 23*(4), 3–13.

Ridgeway, G. (2007). Generalized boosted models: A guide to the gbm package. *Update*, *1*(1), 2007.

Salton, G. & McGill, M. J. (1986). Introduction to modern information retrieval.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*(1), 1–47.

Sherman, L. W. (1993). Defiance, deterrence, and irrelevance: A theory of the criminal sanction. *Journal of Research in Crime and Delinquency*, *30*(4), 445–473.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems* (pp. 2951–2959).

Steffensmeier, D. J., Allan, E. A., Harer, M. D., & Streifel, C. (1989). Age and the distribution of crime. *American journal of Sociology*, *94*(4), 803–831.

Steinberg, L. D., Cauffman, E., & Monahan, K. (2015). *Psychosocial maturity and desistance from crime in a sample of serious juvenile offenders*. US Department of Justice, Office of Justice Programs, Office of Juvenile Justice and Delinquency Prevention Laurel, MD.

Stewart, Z. (2014). Using scikit-learn pipelines and featureunions. Retrieved from http://zacstewart.com/2014/08/05/pipelines-of-featureunions-of-pipelines.html

Șulea, O. M., Zampieri, M., Vela, M., & van Genabith, J. (2017). Predicting the law area and decisions of french supreme court cases. *arXiv preprint arXiv:1708.01681*.

Surden, H. (2014). Machine learning and law. *Washington Law Review*, *89*, 87–116.

Ulmer, J. T. & Steffensmeier, D. (2014). The age and crime relationship: Social variation, social explanations. In *The nurture versus biosocial debate in criminology: On the origins of criminal behavior and criminality* (pp. 377–396). London: SAGE Publications Ltd.

van der Laan, A. & Beerthuizen, M. (2018). Jeugdige strafrechtelijke daders. *Monitor Jeugdcriminaliteit 2017*, 47–68.

van 't Hul, I. (2018). *Improving online trade fraud complaint classification by applying machine learning techniques* (Bachelor's Thesis, Utrecht University).

Varma, S. & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, *7*(1), 91.

Wartna, B., Blom, M., & Tollenaar, N. (2011). The dutch recidivism monitor. https://www.wodc.nl/binaries/the-dutch-recidivism-monitor-2011-20110803_tcm28-78143.pdf, Last accessed on 2019-04-15.

Weijers, I. & Liefaard, T. (2007). Youngsters. In *Dutch prisons* (pp. 127–166). Den Haag: BJU Legal Publishers.

Weiss, G. M. (2009). Data mining in the real world: Experiences, challenges, and recommendations. In *Dmin* (pp. 124–130).

Wiarda-Committee. (1971). *Jeugdbeschermingsrecht*.

Wolpert, D. H. & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67–82.

Xu, L., Krzyzak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, *22*(3), 418–435.

Yang, Y. & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval* (pp. 42–49). ACM.

Yousfi-Monod, M., Farzindar, A., & Lapalme, G. (2010). Supervised machine learning for summarizing legal documents. In *Canadian conference on artificial intelligence* (pp. 51–62). Springer.