Predicting Criminal Cases of Oromia Supreme Court Using Machine Learning

Girma Assefa Woyessa

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

October 2021

Adama, Ethiopia

Predicting Criminal Cases of Oromia Supreme Court Using Machine Learning

Girma Assefa Woyessa

Advisor: Teklu Urgessa (Ph.D)

A Thesis Submitted To
The Department of Computer Science and Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

October 2021
Adama, Ethiopia

Approval Sheet

I/we, the advisors of the thesis entitled "**Predicting Criminal Cases of Oromia Supreme Court Using Machine Learning**" and developed by **Girma Assefa Woyessa**, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Teklu Urgessa (Ph.D)	_____	_____
Major Advisor	Signature	Date
_____	_____	_____
Co-advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis **Girma Assefa Woyessa** have read and evaluated the thesis entitled "**Predicting Criminal Cases of Oromia Supreme Court Using Machine Learning**" and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science Engineering

_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis are contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies, Dean	Signature	Date

Declaration

I hereby declare that this Master Thesis entitled "**Predicting Criminal Cases of Oromia Supreme Court Using Machine Learning**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Girma Assefa Woyessa		
Name of the Student	Signature	Date

Recommendation

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "**Predicting Criminal Cases of Oromia Supreme Court Using Machine Learning**" prepared under my/our guidance by **Girma Assefa Woyessa** submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science Engineering. Therefore, I/we recommend the submission of the revised version of the thesis to the department following the applicable procedures.

Teklu Urgessa (Ph.D)	_____	_____
Major Advisor	Signature	Date

ACKNOWLEDGMENT

First of all, I thank Almighty God with his mother, the Blessed Virgin Mary, who has made me successful in all my academic endeavors, and who has blessed me with perfect health.

My next and most grateful step is to go to my advisor, Dr. Teklu Urgessa (Ph.D.) for a series of support and guidance at this stage of the work. Without his involvement and intervention in difficult stages, it would have been very difficult to complete this thesis according to the initial set plan. It provides important and useful feedback and suggestions on how to approach a research problem tactfully and systematically. Thank you always!!

Next, I would like to express my deepest gratitude to Dr. Wazih Ahmad (Ph.D.) for giving the right direction, useful comments, insightful suggestion and participation from the beginning to the end.

Finally, I would like to thank all the members of the Artificial Intelligence SIG and my classmates for their supportiveness throughout my career.

Finally, I would like to thank my family (Aberu Becha (mother) and Assefa Weyesa (father) for their support and advice in all their endeavors. Especially, my sister Birtukan Assefa for your encouraging and helping me in all directions.

"*Maatii koo isinaan jaaladha umurii dheera naf jiradhaa"*.

Table of Contents

List of Tables

TABLE PAGE

List of Figures

FIGURE PAGE

Acronyms and Abbreviations

A

AI · *Artificial Intelligence*
ANN · *Artificial Neural Network*
AUC · *Area Under Curve*

B

BOW · *Bag of Word*

C

CART· Classification And Regression Trees
CSV · Comma Separated Values
CBOW · Continous Bag of Word
CPU · Centeral Proccesing Unit

D

DT · *Decision Tree*

E

ECHR · European Convention Human Right

F

FDRE· Federal Democratic Republic of Ethiopia
FN· False Negative
FP· False Positive
FPR· False Positive Rate

G

GB · Gradient Boosting

H

HTML · *HyperText Markup Language*
HTTP · *HyperText Transfer Protocol*

I

IDF · *Inverse Document Frequency*

J

JSON · *Javascript Object Notation*

K

KNN · *K-Nearest Neighbor*

M

ML · *Machine Learning*
MS ·

N

NB · *Naïve Bayes*
NLP · *Natural Language Processing*
NLTK · *Natural Language Tool Kit*

O

OCD · *Oromia Cassation Division*
OCR · *Optical Character Recognition*
OLF · *Oromo Liberation Front*
OSC · *Oromia Supreme Court*
OVR · *ONe Vs Rest*

R

ROC · *Reciever Operating Characteristic*

S

SVC · *Support Vector Classifier*
SVM · *Support Vector Machine*

T

TF · Term *Frequency*
TF-IDF · Term *Frequency–Inverse Document Frequency*
TN · True Negative
TP · True Positive
TPR · True Positive Rate

V

VSM · *Vector Space Model*

ABSTRACT

In this world, every victim needs a fair and unbiased judicial decision. A judicial decision is a process of deciding according to the law on the guilty person for the committed crime. This process is carried out by judges. Judges are human beings; they can be inclined into some groups or individuals due to various reasons. In addition to this problems, with human judicial decision-making, there is a low similarity of their judgment among different judges on the same cases. Applying a machine learning prediction model would improve decision-making quality and efficiency by automating the process based on a real dataset. This study aimed to predict a judicial decision of the Oromia Supreme Court (OSC) using machine learning techniques. The judicial decision outcomes would be predicted from two aspects: the judgment (identifying whether the suspect committed the alleged crime or not) and the penalty (if the suspect is found guilty of the alleged crime, impose penalty) using criminal case dataset collected from OSC. The new dataset has 1638 instances that were used to train the models. This dataset hasn't a balanced instance class. The synthetic minority oversampling technique was applied on training dataset and generated 1736 data to balance. An experimental approach was performed in this study to determine best model. Machine learning models based on Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB), with feature extraction techniques like Term Frequency Inverse Document Frequency (TF-IDF), Bag of the Word (BOW), and N-gram have been experimented for both judgment and penalty prediction separately using 10 fold stratified cross-validation. Different classification metrics are used for evaluating models. The RF model with TF-IDF performing best than other models for predict judgment. Since it scored an accuracy of 98.5 % and an f1-score of 98 %. The SVM model with TF-IDF performing best than other models for predict penalty with an accuracy of 79.68 %, and an f1-score of 79 %. Finally, the best-performed model was evaluated by legal experts and achieved 77.5% accuracy. Therefore, RF and SVM models with TF-IDF feature extraction are recommended to effectively predict the judicial decisions of OSC. This study contributes to developing and evaluating machine learning models to predict judicial decisions with penalties. However, it would be better if in depth analysis with all types of law on big corpus will be experimented with deep learning approach for better results.

***Keywords**: Predict Judicial Decision, Ethiopian Criminal Code Procedure, Criminal Law, Machine Learning, and Feature Extraction.*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Courts were established to serve the victim equally by giving unbiased judicial decisions. Before modern laws were enacted in the world, the local courts were presided over by the Lord or one of the pastors. The royal court was initially governed by at least the king himself (Murphy & Plucknett, 1957) (Duncan, 2021). According to the history of the legal system, the first extent law code was formulated by ancient Sumerian ruler Ur-Nammu in 22nd century BC. Then next, King Hammurabi in 1760 BC further develops the Babylonia law by inscribing and codifies it in stone. The birth of the legal system was explicitly started from here and acquired the today name that called judicial decision (Randall Lesaffer, 2009). Judicial decision-making is a legal analysis based on the provisions of the law when resolving a legal dispute, the presentation of the decision to resolve the dispute, and identify the factors that often led to the argument or dispute on the offender and giving the verdict (Pound, 1923).

In Ethiopia, there are no codified laws and written constitutions before 1434. At that time the Ethiopian people were administrated by customary law. Fawse Manfasawi and Fetha Negest were attempted to compile between 1434 and 1930. In-country, the modern codified started in 1931. During this time, Ethiopia used the 1994 constitution after amended the constitution several times. The Constitution of 1994 establishes a federal system of government in which the sovereignty of Ethiopia is nations, nationalities, and peoples. In the country, every court is independent. Government power was initially divided horizontally between the ten autonomous states and the federal government. The Ethiopian legal system categorizes and identifies its legal subjects as labor law, criminal law, business law, family law, tax law, sale law, contract law, tort law, and so on. Those types of laws have their meaning and court proceedings. Of these, criminal law is the body of law that defines criminal offenses and fixes penalties related to crime. Criminal proceedings result in loss of life and liberty. Ethiopia has two judicial systems with two parallel court structures. Each state and the federal government have their supreme, higher, and first instance courts (Abate, 2014).

Oromia state is one of ten member states in Ethiopia and it has Supreme, higher, and first instance courts. In the region, there are 304 first instance courts, 22 higher courts, and 1 Supreme Court. The Oromia Supreme Court (OSC) is responsible for ensuring the unity and predictability of justice services in the region (Oromia Supreme Court, 2019). OSC has one cassation division, it has the full adjudication to review and overturn decisions in the courts of the first instance, higher courts, and the regular division of itself. At this time, the court gives the judgment service through judges (manual) for the community. To accomplish these tasks the court has been hired 2500 judges excluding other staff. However, different levels of knowledge, honesty, and seeing the perspective of judges make it difficult to run the judicial system in the same way at every similar instance of a crime. So, they can lean towards one person or group for different reasons (politics, kinship, and corruption). Further, OSC receives appeals from 18 regional zones. This can also lead to overburdening of judges and increase the time of the trial or sufferings due to manual work.

Computer machines or other technology devices are essential to overcome this and other problems (Greenleaf, 1989). Lawlor (Lawlor, 1963) assumed that computers could one day analyze and predict the outcome of justice. In recent years, to build thriving predictive models of judicial outcomes, the advances of Natural Language Processing (NLP) and Machine Learning (ML) provides the tools and techniques to automatically analyze legal materials (Li et al., 2018).

Many researchers were tried to predicting the legal outcomes using machine learning for the purpose of solving such problems. But they are limited to only judgment prediction (guilty or not guilty, conviction or acquittal) without including penalty or punishments. Without a penalty of guilty, the decision cannot be called a decision, because the punishment of the guilty is necessary for court. That means, if you only predict the judgment without a penalty, that is not called a full judicial decision prediction. And also the way of preparing their dataset is not based on the legal text, but rather by reading repeatedly and identify or extract the factors manually (Marr, 2018).

Our objective will be to help improve the decision-making process at the end of judges, decrease the time of delivery of legal verdicts, and minimize the chance and bias errors in the process. Therefore, this study focused on predicting the outcome of cases from the Oromia Supreme court related to criminal law using machine learning. ML is a subset of

artificial intelligence (AI) and it can be applied to many problems. Machine learning can be supervised, unsupervised, or Semi-unsupervised and reinforcement. The application of machine learning has a great contribution to the modernization of the environment of the court, such as legal document summary, legal aid support, and public legal education. In addition to this, machine learning has the ability (skill) to analyze data to make better predictions of legal outcomes.

In this study, the results of judicial decisions would be predicted from two aspects: the judgment (accusation) and the penalty. To better support and even predict these decisions, we used machine learning techniques to help the computer understand the relationship between this information and the outcome of the trial. Because, predicting legal decisions is important to support non-lawyers, lawyers, and judges to understand what will happen in court proceedings and to improve the quality of their work (Visentin et al., 2019). To our knowledge, this is the first study to forecast judicial decisions in Ethiopia for Oromia Supreme Court.

1.2. Motivation of the Study

According to 2017 population statics[1], in the Oromia region, there is above 35 million populations. But, in OSC there are only 2500 judges. For the 35 million populations 2500 judges are not enough. Due to this technological aid is needed to support courts in order to give efficient judgment service.

The other problem that motivated us is corruption, delay, uncertainty, ill information, lack of coordinated decision making, difficult for a bench of judges to work on the same case. In many courts, corruption is sometimes perpetrated by some judges; this was leading to miscarriages of justice and discrimination. According to an OSC report, only in 2012 and 2013 E.C year, 861 workers were punished for misconduct. In court, the efforts of automating the case and file digitization are still not solved, especially in our country. In addition to these, in one case it takes a long time to acquire a verdict due to work loaded in courts.

On other hand, as described in (Kedia & Rasu, 2020), the prevalence of machine learning in the legal domain is low. Because the legal domain is very difficult to analyze the legal case and needs a legal expert. For this reason, much research remains in the Proof of

[1] https://datacommons.org/place/wikidataId/Q202107

Concept (POC) phase, and adoption is minimal. Therefore, this problem has prompted or motivated us to demonstrate the application of machine learning in the legal domain, especially in the context of Ethiopian law. Generally, this initiates us to design and develop a judicial decision prediction model for OSC.

1.3. Statement of the Problem

OSC is one of the state region courts in Ethiopia and they perform the interpretation and implementation of Ethiopian law. OSC receives an appeal in 18 different higher courts to see the break of the law. The growth in the number of reported cases creates difficulties for the general legal practitioners to cope with the many legal gaps and make timely decisions. This resulted in a significant delay in court proceedings. According to the court statics, in 2011 E.C, 590,452 cases were newly lodged and 49,722 were transferred from 2010 E.C, year, total in region courts 640,174 amount of cases have been seen and only 597,565 cases acquire a full verdict [[2]].

OSC judges and community say there is a big problem between judges from the Supreme Court up to the first instance Court. These problems are, different judges make different judgments on the same cases. That is, sometimes there is no similarity of judgment in the same case between different judges. This problem can be caused by a lack of knowledge, negligence, and relatives.

In many courts, judges can change their decision because of corruption or by looking at the victim's appearance. Corruption is the root cause of injustice and discrimination. Because judges are human beings, they can be ideologically inclined on various issues (politics, reprisal, kinship). The other problem related to OSC is about lawyers. Lawyers provide clients with advice and counsel on their legal rights and obligations. A Lawyer represents companies, individuals, and some groups during legal proceedings and litigation (Vilchyk, 2018). When lawyers represented somebody during litigation sometimes they may be deceiving individuals who are non-law professionals or who do not know the law to make improper profits or money. These and the aforementioned problems are the real problems that OSC encountered.

To solve such problems, a significant amount of researchers have been proposed their work to predict a judicial decision in a different country using the various approaches. However,

[2] https://oromiacourt.org/manneen-murtii-oromiyaa/

besides, predict the first judgment (decision), they aren't predicting the penalty of guilty. That means they can only apply predicting system on the violation or no violation of the articles, affirmed or reversed documents, guilty or not guilty (Strickson & De La Iglesia, 2020) rather than predict the punishments of guilty.

On other hand, as described in (Lage-Freitas et al., 2019)(Loevinger, 1963) several other legal systems in the world share the very same problem of predicting legal decisions to create the best prediction system. This challenge or problem is how to predict legal decisions with a satisfactory level of accuracy to support the work of attorneys, judges, and other professionals. By satisfactory, we mean that the quality of the prediction in terms of accuracy or precision should be better or even higher than Law experts. Nevertheless, it is still very hard to perform any legal decision prediction with adequate accuracy or precision.

1.4. Research Questions

This study addresses the problem by answering the following question:

RQ1: How precisely can the judgments made by the Oromia Supreme Courts be predicted with punishment using machine learning?

RQ2: Which machine learning models and feature extraction are most suitable for judicial decision prediction?

RQ3: How can we improve the quality of judicial decision predictions?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of the study is to predict the judicial decision of the Oromia Supreme Court using a machine learning technique.

1.5.2. Specific Objectives

- To review the literature relating to machine learning concepts to predict judicial decisions.
- To review the Ethiopian Criminal Code Procedure to identify the factors that affect judicial decisions.
- To collect and prepare case document datasets to train and test the classification model.
- To design a prediction model of judicial decisions.

- To find the best machine learning model parameters to optimize model performance.
- To evaluate the performance of the developed model.
- To develop a prototype of judicial decision prediction.

1.6. Scope and Limitations

1.6.1. Scope of the Study

The study focuses on using the machine learning technique to provide a model that predicts judicial decisions for the Oromia Supreme Court. The study was designed to train the dataset on a specific machine learning method appropriate for our study and to use cross-validation techniques and classification metrics for model performance evaluation with various parameters. There is no dataset to train models, so a new dataset needs to be built for this study. Therefore, a dataset is constructed by collecting criminal case documents.

Firstly, the developed model predicts the defendant's guilt or not guilty that called judgment prediction. After reviewing or investigating the judgment prediction result, if a judgment prediction result is guilty, it predicts the penalties or punishment. The way of model working is tried to mimic the actual work of judges.

1.6.2. Limitations of the Study

OSC uses Afan Oromo to carry out judgment services in the region. In short, the data we have collected and prepared is based on the Afan Oromo language. As a result, our work is limited to the Oromia Supreme Court. There are many categories of the Ethiopian legal system and articles that implementation by OSC. In order to limit the scope of this research, we have decided to drop the other categories of cases and chosen criminal cases for our study. The data for criminal cases is radially available and the judgments are also given in less average-case time and number of hearings as compared to civil cases making training data unnecessary complex. Sometimes, the public prosecutor would be accused of the offender (individuals, group, or company) by more than two lawsuits or charges. However, our study only focuses on one lawsuit (sue) at one time. That means we didn’t include more than two sue on one accused person at the same time.

1.7. Significance of the Study

Once the judicial decision predictions and its prototype are fully implemented, it helps judges, lawyers, and the court community to know the outcome of cases. Primarily, OSC benefits from this. Day to day, the applicant of appeal is increasing, for this reason, the burden would be increased on the judges and the litigation takes a long time to obtain a decision. So, after successful completion of this study means based on the attributes given by the user it is very useful in reducing the load of judges, remove the bias of judges, reduce decision time and create the similarity of judgment between judges.

Secondly, attorney or lawyers and non-lawyers benefits from this. Lawyers use this model to analyze lawsuits if their clients could win the lawsuits or not. For instance: Clients often ask their legal advisers or lawyers, "If we go to court, how lucky I am to win or lose", therefore, with this model, lawyers can better answer these questions on time and give confidence to the clients. And also unfamiliar people with legal can simply interact with the model to know the court proceeding and outcome of their results. If we were to encounter a potential victim, we would be able to predict the outcome of their case would be. People who are unfamiliar with the law will find predictions about the outcome of their cases. Generally, the forecasting results can help the judges and lawyers to predict decisions and also help the non-legal professionals to have a basic understanding of the cases.

Thirdly, the corpus prepared for this research will be used for further scientific research. For this study, we went through many challenges to prepare the judgment dataset.

1.8. Organization of the Thesis

In this section, we offer a summary of the chapters covered in our thesis report.

Chapter One of this section deals with the background of the study, the motivation behind the study, a description of the problems with the research question to be raised and addressed, the scope and limitations, the objectives of the study, and its explanation also included.

Chapter two presents the literature review and related works on a judicial decision prediction, the overview of the judicial decision system in Ethiopia as well in OSC, an overview of Afaan Oromo language, Methodologies use in judicial decision prediction:

feature extraction and machine learning classifier by previous researchers, the related work with their comparison greatly discussed and briefly seated.

Chapter three presents research methodologies. In this, we mentioned or put different methods, tools, techniques, and machine learning classification algorithms, model selection techniques that we used in our study.

Chapter four, in this chapter we were, discusses the design and architecture of the proposed solution. PJD model architecture, data preprocessing architecture, feature extraction architecture, and prototype architecture was detail described.

Chapter five describes the implementation and experimentation of the proposed solution and also includes some sample code with their explanation.

In chapter six, the obtained results from experimentation and implementation, dataset class distribution result, feature extraction result, models evaluation results, hyper-parameter tuning results were discussed in detail including machine learning models comparison with and without remove stop-words to figure out the most performing model. It includes also the recommendation and future research directions of this study are clearly stated.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

In this chapter, we examine relevant literature to better understand the concept and to investigate the problem. This provides the technical evaluation of existing techniques used to make judgments and predictions. Starting by defining the meaning of judicial system as world and Ethiopia, and discover an overview of OSC, Afaan Oromo language and followed by a review of predictive modeling, machine learning for legal text classification, and algorithm, and techniques that have been used by previous researchers in the prediction of judicial decision and related work previously studied on the problem.

2.2. The Judicial Decision System

The judiciary system is a system of courts that adjudicates and interprets legal disputes, defends and enforces the law in legal matters. The judiciary generally does not enact or enforce the rule of law, but rather interprets, advocates, and applies facts to every case (Randall Lesaffer, 2009). The judiciary is the branch of government that administers justice according to law. The term is widely used to refer to the courts, magistrates, judges, adjudicators, and other support staff who run the system (Parliamentary Counsel, 2013). The first extent law code was formulated by ancient Sumerian ruler Ur-Nammu in 22nd century BC. Before the codified law was becoming, people in this world would take their cases to the elders or the king.

As the history of the judiciary Romans said, there were the rules of conduct of Mos Maiorum that were based on social norms created over the years by predecessors. They would merely have to judge the case rather than the others. In some countries, however, the justice system has its own rules [3].

2.3. Judicial System of Ethiopia

In Ethiopia, before 1434 there are no codified laws and written constitutions. However the modern codified law started in 1931. Before modern laws were enacted in Ethiopia, people living in various regions used to take their cases to their local elders, commonly known as "Jarsolii", acting as arbiters. In the context of Ethiopian traditional society, the word

[3] https://en.wikipedia.org/wiki/Judiciary

"case" represents the Oromo word "Falmi", which means a dispute between two parties, brought before the attention of a certain judicial body seeking a decision from it. Since there were no prepared courtrooms in place at the time, the trial was held under large trees in the area. Such places are now commonly referred to as "Dhaddacha," which means a place where any section of the community can come and observe the process.

Ethiopia has two judicial systems with two parallel court structures: state courts and federal courts with their individual or own administrations and independent structures. Judicial power is vested in the courts at both the state and federal levels. The Federal Supreme Court includes the Cassation Division, which has the power to examine and overturn decisions made by the lower federal courts, the Supreme Court itself, the regular division, and the state Supreme Court. In addition, the decisions of the Federal Supreme Court's Cassation Court on the interpretation of laws apply to both federal and state courts [4]. Each state has a first instance, a higher court, and a Supreme Court. There are also local Sharia courts dealing with Muslims on religious and family matters. The Federal Supreme Court and the Federal High Court have jurisdiction over federal law, cross-border cases, and domestic affairs. The constitution provides for an independent judiciary. Defendants have the right to legal counsel, and the public defender's office, like a university, other organizations, etc. provides counseling for incompetent defendants (Abate, 2014) [5] [6].

The Ethiopian legal system categorizes into labor law, criminal law, business law, family law, tax law, sale law, contract law, tort law, and so on. Criminal law, in contrast to civil law, is a legal system that focuses on punishing those who commit crimes. In most cases, the legislature establishes criminal offenses and punishes them. However, criminal law varies depending on the authority (Suddarth & Koor, 2018).

2.4. Overview of Oromia Supreme Court

The FDRE constitution offers for the establishment of three (3) levels of government courts. These are called federal supreme courts, federal high courts, and federal first instance courts. The Oromia Supreme Court also has this structure.

[4] Federal Courts Proclamation Re-Amendment Proclamation 454/2005, Article 2(1).
[5] https://www.nationsencyclopedia.com/knowledge/Ethiopia.html
[6] Federal Democratic Republic of Ethiopia [FDRE] Constitution Art. 8(1).

The Oromia Supreme Court is one of the regional courts in Ethiopia. State courts have a general mandate to provide impartial accessible, efficient, and effective judicial services. They carry out their duties freely and with great responsibility. OSC has its own hierarchy to provide access to justice for the community. It has been categorized into First instance courts (woreda), zone high courts, and Supreme Court. In the first court, the first courts have jurisdiction over minor cases to making a decision. Each decision, in the first courts, is judged by only one judge. If a person or some group does not believe in this decision, they have the right to appeal to the zonal high court.

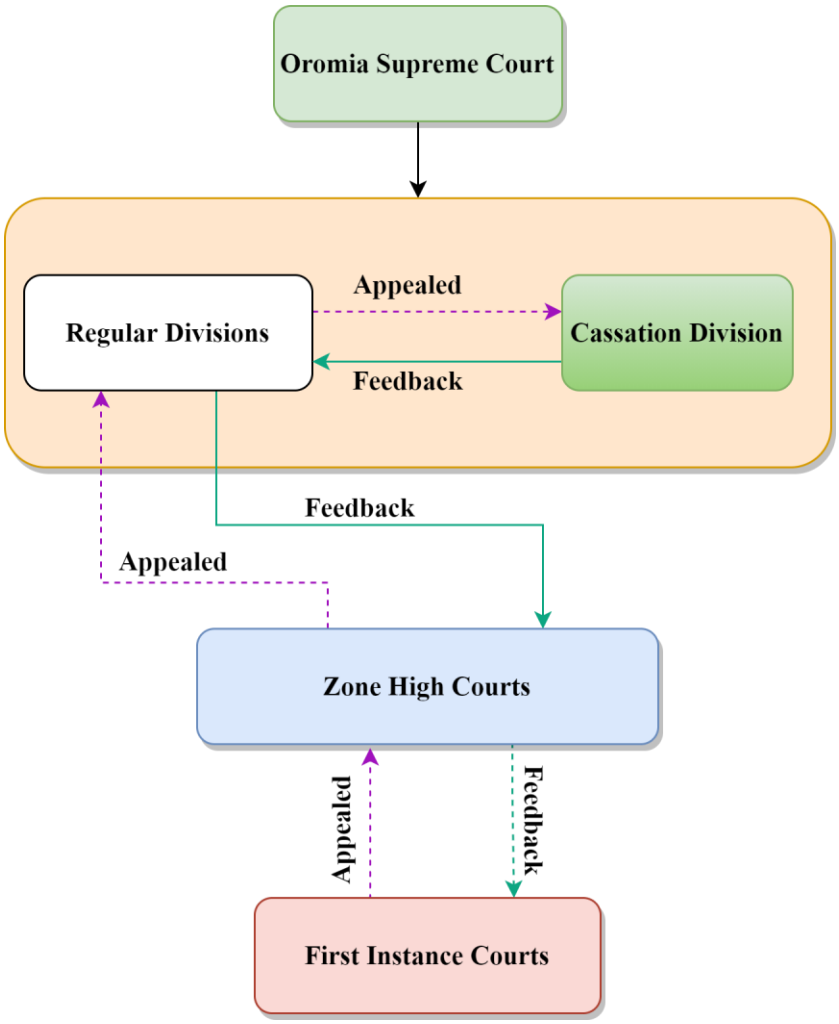

Figure 2.1: Oromia supreme court judicial decision structure

In zone high court, the high courts also have jurisdiction that is greater than the first court to make decisions including appealed cases. It also has the power to order and administer the first courts. Every decision, in the high court, is judged by at least three judges. In the Supreme Court, there are two divisions of judges. Those are regular division and cassation

division. Regular divisions have a jurisdiction to see and review the case that appealed to the Supreme Court from different high courts and the new cases opened at Supreme Court. The regular division is appointed by three to five judges. However, the Cassation division Inquiry investigates fundamental legal errors in the region. It is made up of seven judges, including the president and vice president of the Supreme Court. Their verdict is used as a law for similar cases in the Oromia region.

In OSC courts, there are some technology results like court management database, sound recorder machine, queue management system that used for making efficient judgment service in courts.

2.4.1. Judicial Process in OSC

There are processes in the courts that resolve disputes between two or more people, this is called court procedure (Abdi, 2016). There are different types of court proceedings depending on the case. Of these, the criminal procedure code of Ethiopia is one. According to (Imperial Ethiopian Government, 1969), criminal procedure controls the entire process of detection, investigating, prosecuting, and punishing offenders. It has two functions. It enforces criminal law. At the same time, it helps to protect the rights of the suspect or defendant. In addition, in contrast to civil cases, criminal proceedings can result in loss of life and liberty. However, the court proceedings code includes the process of how judges give a judgment against the offenders, how the prosecutor prepares charges, and how the defendant defends his or her rights.

In generally a judicial decision made by OSC it must contain the following main parts (Imperial Ethiopian Government, 1969):

Introduction, consisting of the judge's name, a plaintiff (appealed) and defendant name (respondent), date, and certain description of the case.

Judgment, consisting of the following parts:

Lawsuit: containing relevant background information on the applicant and the events and circumstances that led them to seek justice due to alleged violation of their rights according to the criminal law of Ethiopia. And also contains a relevant provision from legal documents (relevant law).

Law, containing legal arguments of the Court with each alleged violation.

The word of faith, after the case has been presented to the court and has been read to the accused, the court will ask the defendant whether he/she/they committed the crime or not.

Witness, there are two witnesses according to criminal code procedure. Those are prosecutor witnesses and defense witnesses. Prosecutor witness is the written or human evidence that proving the committed crime. Whereas, defense witness is the written or human evidence that is defense from the accused.

Appeal, after a final decision has been made, either party or both may appeal from the judgment if they believe there had been a procedural error made by the trial court. So, containing some description of why the appeal would be taken.

Decision, after examining the case under Ethiopian law, the court will decide whether the defendant is guilty or not.

Verdict, after the court decided the accused person is guilty according to the lawsuit; the punishment or verdict would be implemented based on the idea of mitigating punishment raised by defendants and the idea of increasing the punishment raised by an attorney.

Command (order), finally, after the judicial decision is made, the judges give different orders. For instance, if the defendant is found guilty and punished, the court will write the order to the prison commission to enforce the sentence or if the defendant is not guilty, the court will order his release from prison. In addition to this, the court gives different commands.

2.5. Overview of Afaan Oromo Language

In this study, the Afan Oromo language has been discussed. This is because the Oromia Supreme Court has been using Afan Oromo as a working language since 1991, including the High Court and the First Instance Court. All cases that mean the lawsuit, an answer of the accused, witness term, etc. including judicial procedure are performed by Afaan Oromo language.

Afan Oromo is an Afro-Asiatic language and is widely spoken in Cushitic subtitles. Oromo is widely (extensively) used in both written and spoken languages in Ethiopia and certain neighboring countries, including Somali and Kenya, and is spoken as (Asafa, 2010) by more than 40% of the population in Ethiopia. Afan Oromo is the working language of the Oromia regional government and the language of instruction in primary and secondary

schools throughout the region as well as the working language of the public and private offices.

The Oromo people have not developed an independent writing system, so, they use a highly developed oral culture. In the 19th century, scholars began using the Latin alphabet to write in Oromo [7]. In 1842, John Ludwig Krapf began translating the Gospels of John and Matthew into Oromo, as well as the original grammar and vocabulary. The first Oromo dictionary and grammar were compiled in 1844 by German scholar Karl Tuchech.

Oromo was written officially in 1991 in the form of the Latin alphabet, QUBEE [8]. Various Latin-based versions were previously used by the OLF and by Oromos outside Ethiopia in the late 1970s (Negesse, 2015). It is believed that more text was written by Oromoo between 1991 and 1997 during the adoption of QUBEE.

2.5.1. Afaan Oromo Alphabets and Orthography

Afan Oromo script is called "QUBEE". It uses the Latin alphabet or characters (Roman orthography) in writing. Like Amharic, Omotic, Tigragn including English and other languages, Afan Oromo has several vowels and consonants. Afaan Oromo vowels are five in number. Those are A, E, I, O, and U. Afan Oromo has 27 consonants, including combined consonants (CH, DH, NY, SH, and PH) called "**Qubee Dachaa**". The letters "Z", "P", and "V" are not Afan Oromo letters because none of them are written in Oromo. Generally, as depicted in table 2.1, the Afaan Oromo alphabet is thirty-two (32) in number (Wikipedia, 2021).

Table 2.1: Afaan Oromo alphabets [9]

Aa	Bb	Cc	CHch	Dd	DHdh	Ee	Ff	Gg	Hh	Ii
Jj	Kk	Ll	Mm	Nn	NYny	Oo	Pp	PHph	Qq	Rr
Ss	SHsh	Tt	TSts	Uu	Vv	Ww	Xx	Yy	Zz	

[7] https://en.wikipedia.org/wiki/Oromo_language
[8] "Afaan Oromo". University of Pennsylvania, School of African Studies.
[9] https://www.africa.upenn.edu/Hornet/Afaan_Oromo_19777.html

2.5.2. The Writing System of the Afaan Oromo Language

The writing system, technically known as script or orthography, consists of a set of visual elements, shapes, or characters, or graphs that contain certain structures in a language system. We use different writing systems with symbols representing different things: logographic, syllabic, and alphabetic. In alphabetical order, a symbol connects to a sound in the language, rather than to an object, an idea, or a word.

Each language has its own rules and syntax. In the English alphabet, two letters have different purposes. Consonants include most English letters. Vowels are small in number and can be combined with vowels to make different sounds. However, the main difference between the Oromo and English spelling systems is how a word is spelled. There are no rules for how a word is written or spoken in English and you have to remember the word in your heart, but there are rules which are called "**Serluga**" Afan Oromo that anyone can follow to write and read the Oromo words (Duresa, 2016).

a) Spelling Rules

According to (wikibooks, 2021), "Qubee" replaced the various Oromo alphabets with Latin alphabets, helping to make the Oromo alphabet standard. Spelling differences still occur due to personal preferences and speech differences. In Oromo, a word cannot begin or end with two consonants; most Oromo words end in a vowel. The word "defendant" has been changed to "himatama". In Afaan Oromo, misplaced words make the spelling completely different.

There are four rules for writing Afan Oromo words: 'Sagalee Lafaa' or 'Sagalee Jabaa' and 'Sagalee Dheeraa' or 'Sagalee Gabaabaa'. 'Laafaa' and 'Jabaa' are based on the number of consonants. When the same consonants are written to each other, that sound is emphasized ("Sagalee Jabaa", in Afan Oromo). For example, "**Gubbaa**" means on top. If a single consonant is used, it is "Sagalee Laafaa". For example, "**Gubaa**" means burn. "Dheeraa" and "Gababaa" are conjunctions with repeated vowels. When vowels are repeated, when sounds are stretched or lengthened, it is called "sagalee dheeraa". For example, "**Boonaa**" means proud man. Otherwise, the sound is short which is called, "Sagale Gabaabaa". For example, "**lafa**" means ground.

When two different non-digraph consonants appear in a series, it is called "Irrabuta", a phoneme that is uttered for a very tiny time. For example, Arbaa, Garbaa, daldalaa,

etc., in these words there is a phone [‘I’ or 'i'] in between them which uttered for short time. But, not written. Vowels cannot be changed without rest, and there is a consonant called "Hudha" among them. Spelling preferences and accents may vary depending on what breaks are used. For example, “*very*” can be baa'yee, baayee, baa'ee, or baay'ee, and “*to hear*” can be dhagahuu or dhaga'uu. The diacritical marker indicates the vowels are produced separately, not as a diphthong.

b) Grammar

According to (Thompson, 2021), the Oromo grammatical system is very complex and exhibits many characteristics common to other Cushitic languages, namely, it is a distorted language that uses postpositions rather than prepositions.

Nouns and adjectives: Most Afaan Oromo adjectives and nouns are marked for female or male. When biological sex is associated with a particular suffix, names have a natural masculine or feminine gender that cannot be determined by the form of the noun, such as, **-eessa** for masculine and **-eetti** for feminine nouns, for example, **obboleessa** “brother” and **obboleetti** “sister”. All adjectives and nouns are marked to number plural and singular. For example: for female nouns ‘haroo’ (lake) – ‘harittii’ (the lake) and for male nouns ‘nama’ (man) – ‘namticha’ (the man).

Pronouns: Afana Oromo pronouns have a person (3rd, 2nd, 1st), case (locative, nominative, ablative, genitive, accusative, dative, instrumental), number (plural and singular), and so on. There is differences between near and far display pronouns, for example, ‘kana’ (this) and ‘san’ (that).

Verbs: Oromo verbs consist of suffixes plus the stem, which represents person, gender, number, tense-aspect, mood, and voice. Verbs agree with their subject of person and number. Verbs, except the verb "be", agree with their subject in gender, when the subject is the third person singular pronoun “he” or “she”. In Afaan Oromo verbs there are four moods (indicative, interrogative, imperative, and jussive) and three sounds (active, passive, and the so-called auto benefactive (semi-passive/middle)).

2.6. Application of AI in Law

Artificial intelligence is the mimics of the behavior of human processes in machines, especially computer systems. AI is important because it can perform more difficult tasks than humans. Artificial intelligence tools often complete tasks, especially when it comes to analyzing multiple legal documents to confirm that relevant, detailed tasks are filled in the

relevant fields correctly [[10]]. AI has many benefits and roles in various areas of work, especially in the legal area (Walton, 2005).

There are rapid advances in Artificial Intelligence (AI), which will have significant implications for both the legal profession and certain areas of law (Ashley, 2017). The AI has been applied to the legal or law field to improve the various problems that the legal profession has faced over the years and also it is now being implemented.

AI application in law:

Legal Adviser Support: In this case, lawyers or other professionals may request a system, then a system will review the relevant law stored in its system, gathers and give a response highly relevant to the answer. The US law firm Baker is using ROSS, which a legal adviser system is built by IBM Watson.

Case Outcome Prediction: The goal of most research in AI and law is to develop a computational model of legal reasoning that can give legal arguments and predict the outcome of legal disputes. They do so often use case-based reasoning (CBR) models or ML algorithms, sometimes combining the two (Ashley, 2017).

Question and Answering Chatbots: using NLP, the system responds to the user according to their request. The potential of machines to understand human language and our ability to anticipate our needs is leading to machine-to-human interaction (Partner et al., 2018).

Public legal education: the aim of this is to help ordinary people understand the legally complex problems through teaching, advising, and using another method. AI is the combination of machine learning, deep learning, natural language processing, and more (Partner et al., 2018).

2.7. Machine Learning Algorithm in Law

Other AI and Legal approaches to forecasting, machine learning, uses algorithms for learning from data and use what they learn to make predictions. They use statistical methods to trigger a forecast model (or function) from a dataset to predict the outcome of a new case.

[10] https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence

Machine learning algorithms are programs that can learn from data and improve from experience, without human intervention. Learning activities can include learning structure hidden in unlabeled data, learning the function that mapped to the output from input, Or 'instance-based learning', where a class label (tag) is produced for a new row (instance) via comparing the new row (instance) to row (instances) from the training dataset, which that stored in memory. In terms of classifiers, machine learning approaches can be categorized as supervised, unsupervised, or semi-supervised and reinforcement approaches.

Supervised Machine Learning (SL) (Liu & Wu, 2012): is an ML paradigm for obtaining the input-output relationship information from a system based on a set of combined input-output training samples. Since the result is considered as the label of the input data, a sample of input-output training is also called supervised data or labeled training data. The goal of SL is to construct an artificial system or model that can learn the mapping work between the input and the output. It can predict the outputs (result) of the model or system based on given new inputs. If the output accepts limited discrete values that show the class labels of the input, the learned map will be directed to the classification of the input data. If the output accepts or takes continuous values, it leads data input to a regression.

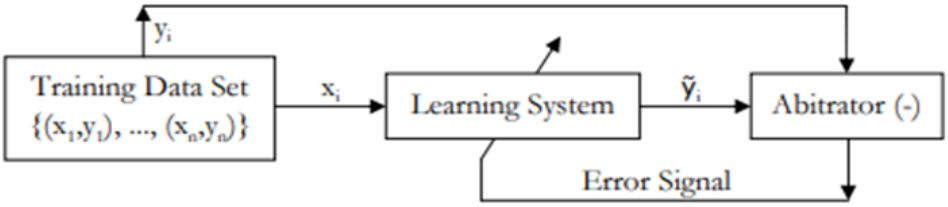

Figure 2.2: Block diagram of supervised learning

There are various supervised machine learning algorithms used in the past that have been used by a particular researcher. Of those, we discuss certain algorithms, because our study focuses on prediction.

Support Vector Machine: (Adrian Erasmus, 2010) SVM is a supervised algorithm. In this algorithm, the data points are separated into a class with hyper-planes. This hyper-plane is built with the help of the margins of each section created from the support vector. A hyper-plane is constructed using kernel trick to indicate that the problem described is nonlinear alignment or linear alignment. The SVM algorithm gives good results in large data sets as well as in large feature areas.

$$w * x + b = 0 \qquad (2.1)$$

Another SVM algorithm is a multiclass SVM, which is used as a classifier on a database that contains more than one component (grouping or category). SVM has been successfully used in many applications such as image detection, diagnostic testing, and text analysis. This algorithm was used by many researchers in the law domain.

Naive Bayes: (Berrar, 2018) Naive Bayes is a probabilistic classification algorithm that uses the Bayes theorem of probability or Bayes laws. The naïve Bayes algorithm is called naïve because it assumes that the occurrence of a particular behavior is independent of the occurrence of other behaviors and that the classification uses Bayes's rule of law. This algorithm predicts membership opportunities for each class, for example, the given record or data point gives the probability of belonging to a specific class. The class which has the highest probability is considered the opportunity class. This is also known as Maximum a Posteriori (MAP).

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \qquad (2.2)$$

Logistic Regression: The logistics regression algorithm (Vidhya, 2015) can be used for both classifications as well as for regression. The logistic regression is similar to linear regression with logistical output. An Independent feature of the data to predict or classify the dependent target value was used. This algorithm is similar to the perceptron neural algorithm where we update the weights to obtain the predicted output to a certain extent.

$$y = b0 + b1 * x1 + b2 * x2 + \cdots + bn * xn \qquad (2.3)$$

Of those, the independent variable (x1,x2,…, xn) are the features of the data, and b0,b1,b2,…, bn are the weights that are updated and y is the output (dependent variable).

Unsupervised Machine Learning: When data are unlabeled or categorized, supervised learning (SL) is not possible, and an uncontrolled learning approach is needed, which attempts to gather natural clustering of the data to groups or clusters and then map new data to these established groups.

2.8. Feature Extraction Methods used in Legal Text

As mentioned in (Waykole & Thakare, 2018), text feature extraction is the process of extracting a list of words from the text data and converting them into a feature set that can

be used by the classifier. This work emphasizes the review of available behavioral techniques. The following information can be used to extract text information.

2.8.1. Bag of Words

A BOW is a way of extracting features used for modeling from the text, such as with machine learning (ML) algorithms. This approach is so simple and flexible that it can be used in countless ways to extract features from documents. And also it is a representation of text that expresses the occurrence of words in a document. It includes two things; those are a vocabulary of familiar or known words and a measure of the presence of known words [[11]]. The shortcomings of this approach are that the order of the word and the integrated and semantic content is ignored. Thus, if the words are used in different contexts, it can lead to incorrect placement.

$$BoW3 = BoW1 + BoW2 \qquad (2.4)^{12}$$

2.8.2. TF-IDF

TF-IDF is a measure of the importance of a word in a document in the dataset and increases when a word appears in the document. The TF-IDF is made up of two words: The first computes the modified Term Frequency (TF), which is the total number of words or terms in that document is divided by the number of times a word or terms appears in a document;

$$TF(t) = \frac{number\ of\ time\ term\ t\ appears\ in\ a\ document}{total\ numbers\ of\ terms\ in\ the\ document} \qquad (2.5)^{13}$$

The second term is the **Inverse Document Frequency (IDF),** which is calculated by dividing the number of documents in the corpus by the number of documents in which a particular word appears.

$$\mathbf{IDF(t)} = \log\_e \frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it} \qquad (2.6)^{14}$$

[11] https://machinelearningmastery.com/gentle-introduction-bag-words-model
[12] https://en.wikipedia.org/wiki/Bag-of-words_model
[13] https://monkeylearn.com/blog/what-is-tf-idf/
[14] http://www.tfidf.com/#:~:text=TF

It is different from a bag of words or N-gram because the frequency of the word is adjusted by the frequency of the words in the corpus, which makes the fact that some words appear in general.

2.8.3. Word2Vec

Word2Vec (Rong, 2014) is used to construct word embedding's. Models created using Word2vec are two-layer neural networks with shallow meanings. Once trained, they reproduce or duplicate semantic contexts of words. The model takes up a large piece of text as input. It then creates a vector space, usually hundreds of dimensions. In the corpus, each unique word is assigned to the corresponding vector in the space. Words with common or same contexts are placed or put nearby in a vector space. Word2vec can use one of the two architectures: a continuous bag of words (CBOW) or continuous skip-gram (CSG). In the continuous skip-gram (CSG), the current word is considered to predict the context next window. In this architecture, adjacent contexts are considered more difficult than words with a distant context. In the continuous bag of words (CBOW) architecture, the order of the context does not affect the prediction, as it is based on the bag of words (BOW) model.

2.8.4. N-Gram

N-grams (Cavnar & Trenkle, 2001) is a word prediction model using probabilistic methods to predict the next word after observing N-1 words in a text. This feature extraction approach consists in combining sequential words into lists with size N. Simply; N-grams are all combinations of adjacent words or characters of size N that are found in the text in a document. This method allows improving classifiers' performance than BOW because it incorporates, to some degree, the context of each word. Rather than using words, it is also possible to use N-grams with syllables or characters. To be more predictable, Character N-gram features have been proved more than the token N-gram features. N-grams are one of the most used techniques in judicial decision prediction and related tasks.

2.9. Related Works

The possibility of applying the NLP technique with machine learning to the legal profession is a very promising and lucrative prospect, and a lot of research is being done in this area using many techniques. Due to the wide range of legal documents, the ML and NLP combination can play an important role in this area for lawyers to provide the necessary information or to ensure that they are accurate. However, as a whole most solutions to date remain in the Proof of Concept (POC) phase, and adoption is minimal (Kedia & Rasu, 2020) in courts. Machine learning also not gave the explication expected from it for the legal domain. Because it is very difficult to extract features in the legal domain, it requires a legal expert. In this section, different related works have been conducted locally and globally that are done by different researchers in the legal domain using machine-learning techniques that are closely related to judicial decision prediction have been reviewed.

Eskinder M., (Eskinder Mesfin, 2009) presented the forecast model to the Ethiopia Federal Supreme Court on active and pending cases. It focuses primarily on predicting the length of time it will take from the date the case is filed to the time the case is decided. This is not related to judicial prediction, which means that the researcher is not focused on judicial decisions. However, the researcher used the ANN model, which contains 9 inputs on 33,000 records. Through these models, the researcher acquired 94.4% of accuracy. Nevertheless, this study, the study which has been done in Ethiopia that related to law.

Nikolas Aletras et al., (Aletras et al., 2016) worked on "*Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective*". In this study, the researcher has conducted the experiments for the prediction task of judicial decisions of the European Court of Human Rights, based on the data obtained from the "HUDOC" online web repository using only support vector machine learning technique. The researchers used a small size dataset. In this study, the researcher determined only two classes (violation or no violation). By the way, the researcher trained the model per article (article 3, 6, 8) and used a balanced dataset of cases (violation and no violation) for those articles. The authors have been applied 10-fold cross-validation only once. The author dividing a judgment into different sections such as procedure, law, alleged violation articles, the facts to make experiments. Finally, the author says; '*we observe that Circumstances is the best subsection to predict the decisions for cases in Articles 6 and 8'*. That means good prediction results were obtained when they using the circumstance

section as features. However, this approach is highly dependent on the quality of data collected and analyzed rather than follow the court's procedure.

Masha Medvedeva (Medvedeva et al., 2020), proposed "Using machine learning to predict decisions of the European Court of Human Rights". The study focused on the work of Nikolas el., This means that they have tried to improve the work of the author cited in (Aletras et al., 2016), with a lot of datasets using the same methods. However, apart from class segregation (violation and non-violation) and adding many data, they did not focus on the defendant's punishment.

Rafe Athar (Shaikh et al., 2020), an author has proposed predicting outcomes of legal cases for Delhi district court. CART, Random forest, KNN & another algorithm is introduced. In brief, CART is performing best in terms of both Accuracy and F1 Score as the researcher said. The study has discussed the outcome of legal cases based on the type of evidence and ideological direction of the court from 86 cases that related to the murder. However, the authors identify or extracted the feature by repeatedly reading a judgment case, and then, collect the important features or factors that affect the outcomes. Alive, evidence, motive, and other 16 features are the features extracted manually. The researcher has mainly focused on binary classification (Acquittal or conviction) without punishment of the guilty person.

Jijajing Li el., (Li et al., 2018), proposed "*A Markov Logic Networks Based Method to Predict Judicial Decisions of Divorce Cases*". The researcher focused primarily on semantic legal factors, which are described based on generative grammar rather than contiguous word sequence. In this article, legal factors come from judgment documents through a knowledge extraction engine. There are many concepts and relationships in the description of legal reasons, so they used generative grammar and a set of context computing operators (and, or, not, etc.) to describe those concepts and their relationships. The researcher collected data on around fifty thousand (50,000) divorce cases from the China judgments online repository. The collected data or document was divided into facts, articles, and decisions. They acquired an F1 score of 77.74% for granted and of 73.58% for dismiss. Nevertheless, the researcher is limited to binary classification (granted or dismiss).

Michael Benedict L. Virtucio (Virtucio et al., 2018) worked on "*Predicting Decisions Using Machine Learning and Natural Language Processing of the Philippine Supreme Court*". The researcher tried to classify the case as affirmed or reversed for Philippine

Supreme Court. To do this work the researcher uses the machine learning algorithm called SVM and also to represents the case text into n-grams the researcher used a BOW model from the NLP technique. Based on the dataset obtained from the online repository they acquired 62% of accuracy. The researchers did not follow the standard format in writing decisions when preparing a dataset. Moreover, they were not imitating the correct court proceedings and they were mainly focused on the affirmed or reversed cases.

Summary of Related Work

Generally, many researchers were tried to make a convenient environment for courts in different countries especially related to judgment. Many of them were focused on only binary classification (violation or no violation, acquittal or conviction, affirmed or reversed) but did not focus on the punishment of the guilty person. Without a sentence, the decision cannot be called a decision, because the punishment of the guilty is necessary for court. Their way of preparing data has not been followed or prepared in accordance with court procedures. In the court procedure, there are known issues that affect or change the decision. Only a handful of researchers have defined and verified the accuracy of the experiment and the cross-validation evidence.

Table 2.2: Summary of related work.

S/N	Title	Author	Gap	Algorithm
1	Application of multilayer feed-forward ANN perception in prediction of court cases time span	Eskendir Mesfin (2010)	• The researcher cannot predict the judicial decision, they only predict the time span of court	ANN
2	Predicting a judicial decision of the ECHR NLP perspective	Nikolas Alteras el (2016)	• Small data sizes are used • They determined only two classes (violation or no violation) • They did not predict the penalty • They trained the model per article • They didn't follow court procedure	SVM
3	Predicting a judicial decision of the ECHR NLP perspective	Medvedeva (2018)	• The researcher use the only 2 classes • They not predict the penalty/ verdict/ after they identify the class	SVM
4	Predicting Decisions of the Philippine Supreme Court Using Natural Language Processing and	L. Virtucio (2017)	• They did not predict the penalty/ verdict/ of cases • They determined only two classes (affirmed or reversed) • They use BOW models	SVM

	Machine Learning			
5	Predicting Outcomes of Legal Cases based on Legal Factors using Classifiers	Rafe Athar Shaikha (2019)	• Small data sizes are used • Only focused on binary classification (Acquittal or conviction) • Extract feature by reading and analyzing case	CART
6	A Markov Logic Networks Based Method to Predict Judicial Decisions of Divorce Cases	Jijajing Li et el. (2018)	• Limited to binary class. • They do not predict the penalty/ verdict/ after they identify the class	Markov logic network

CHAPTER THREE

3. RESEARCH METHODOLOGIES

3.1. Introduction

As mentioned in chapter one, the objective of this study is to predict judicial decisions of OSC using machine learning. In this chapter, we have discussed how and in what ways we have achieved this objective. This includes ways of collecting, processing, tools, or materials used in collecting data, method selection of the model and feature extraction, and research tools. Additionally, it covers systematic methods, practices, and approaches to conduct predicting judicial decisions using machine learning, start from identified research area up to deploy a model.

3.2. Research Design

Research design is a plan designed to answer research questions. So, in order to answer our research questions, we have been designed our research plan as depicted in Figure 3.1. There are a variety of research design approaches, such as qualitative, quantitative, and participatory. This study followed or applied an experimental or quantitative research design approach, which included identifying research objectives and building machine learning models to validate the concept. Quantitative research design approaches find the relationship between two variables i.e., independent variables and dependent variables in data samples (Kamiri & Mariga, 2021).

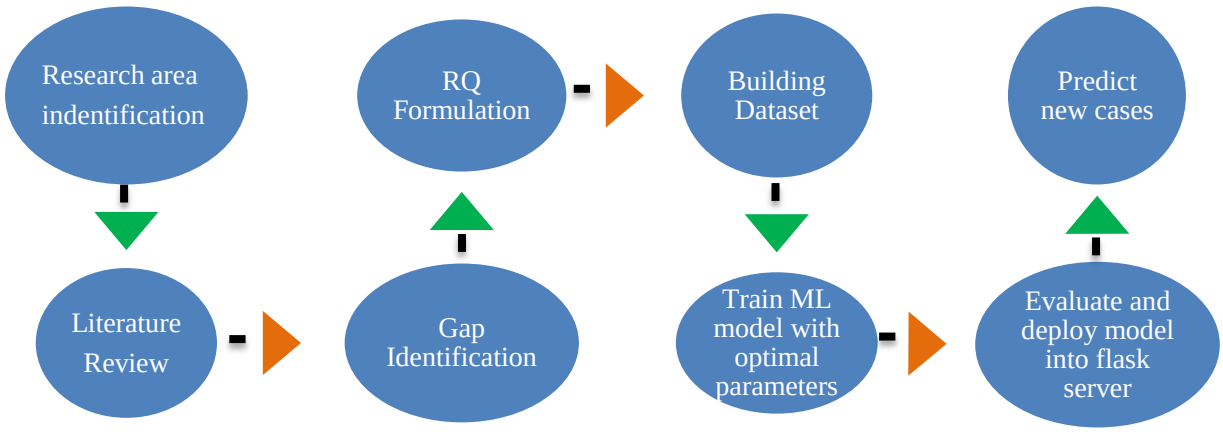

Figure 3.1: Research design procedure of PJD

3.3. Building Dataset

The legal domain or law is too complex and too broad to interpret even by judges. Due to its complexity, there is no dataset and pre-trained models on Ethiopian law. As the case regarding dataset scarcity in studying predicting judicial decisions for OSC is not different from the above-mentioned problem, it's needed to build a new dataset, because there is no published or prepared dataset for this purpose. The process of building the dataset to predict OSC judgments consists of three main stages. Those are:

a. Collecting or gathering case documents and selecting appropriate criminal cases.
b. Preparing, filtering, or consolidating, and converting an image or scanned document to text gathered data into one file dataset.
c. Store the dataset.

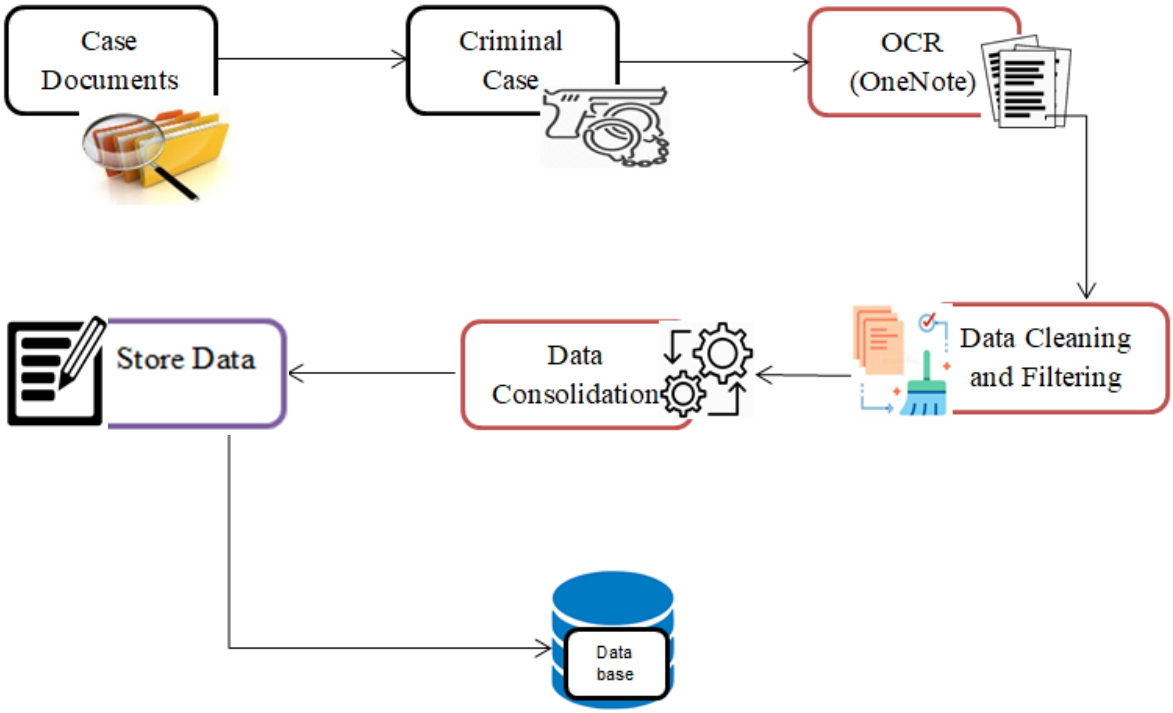

Figure 3.2: Method of building datasets

3.3.1. Data Source

The data to conduct this research was collected from Oromia Supreme Court from a closed case. The OSC is one of the largest regional courts in Ethiopia, with more than 35 million people living in the region, and many cases are dealt with. OSC receives an appealed case from different regional zones or opens a new case at the regional level. After that case acquired a verdict, it will be returned to the archive. This verdict is the verdict made by the cassation division and regular court divisions. Beginning in 2008 E.C., the court began to

convert closed cases from hard copy to soft copy through a scanning machine. So far, different cases documents have been scanned and kept as a soft copy in the OSC court. In general, the judgment of the Oromia Supreme Court was used for our study.

3.3.2. Data Collection

Data Collection is the process of gathering the required information from the applicable sources to provide the desired solution to the existing problem. Machine learning is a data-driven model; the data collection and understanding of the model's features are significant. There are several types of data collection methods, such as primary and secondary data collection, but only the secondary data collection methods are employed in this study.

a) Primary Data

Primary data is a type that is collected directly from an existing source without going through any existing sources. It is mostly collected for other research purposes or projects or and can be publicly shared for other research purposes.

b) Secondary Data

Secondary data is data that has previously been collected by someone else but is for use by others. In this study, various cassation division and regular division case judgments were collected from the Oromia Supreme Court registry. All collected case documents are in PDF format, but its image combined as a pdf format. In all, we have collected more than 8,000 case documents. Those collected case documents have different types of cases, such as civil cases, criminal cases, labor cases, and mixed (tax case, torture case…) cases. For this study, a criminal case document was used. A criminal case is a case that includes endangered, harm, or otherwise endangers people's property, health, safety, and moral security. That means it has to do with human life and freedom. Criminal case documents cover a variety of crimes, including judgment and punishment.

Table 3.1: Initial distribution of cases obtained from OSC

#	Case Type	# of case	Description
1	Civil case	3000	Not selected
2	Criminal case	2000	Selected
3	Labor case	800	Not selected
4	Others (mixed)	2500	Not selected
5	Total	8300	

3.3.3. Data Preparation

In the real world, raw data and images are often incomplete, inconsistent, and lack specific characteristics or tendencies. They are also more likely to make mistakes. Therefore, once collected, data needs to be cleared and prepared. This is because the collected data is scanned or a PDF form. To do this:

First, we classified the case documents as either other or criminal cases. This is done by opening each case document and looking at the type of case on the cover page. We will then keep or put all selected or filtered criminal documents in one folder. The criminal case documents related to murder and injury of the body are selected by looking at the type of accusation on the cover page and placed in each own folder.

Second, the collected data has been improperly formatted, which means, the scanned document is seated in the form of a pdf file, so it's very challenging or impossible to simply extracted or copy the needed text. To handle these issues, Microsoft OneNote was used as optical character recognition (OCR) software. Therefore, using OneNote, we have transformed or changed fact or sue text and decision text into standard or normal text.

Third, after changed the image text or pdf text file to normal text, the following point was implemented,

- Entries that miss some fields (not full case documents).
- Poorly formatted entries.
- Duplicated entries.
- Non-criminal cases

As we described in table 3.1, we have obtained 2000 criminal case documents from OSC. However, we only got 1638 case documents out of 2000 case documents. This is because some case documents do not have complete information, some are not visible or readable by the OCR, and some are duplicated and different offenses. Finally, we have saved the prepared data in excel in the form of tabular.

Table 3.2: The amount of data filtered and prepared

#	Offence Category	# Issue	Case type	Description
1	Body Injury	802	Criminal	
2	Murder	836	Criminal	
3	**Total**	**1638**		

3.3.4. Data Preprocessing Techniques

Whenever data is collected from a variety of sources, it is collected in a raw format that cannot be feasible to analyze. Data pre-processing method is needed for cleaning the case or lawsuit based on basic text pre-processing techniques. It is a method of converting raw data into a clean dataset (J, 2020). This method is used to prepare the dataset for feature extraction. In this study, the following methods are used in pre-processing procedures to make them more suitable for ML.

- Cleaning: remove special character, punctuation, extra space from our dataset and converting it to lower case.
- Make normalization: to reduce the randomness word by bringing it to closer a predefined standard or norm.
- Remove stop-words: to focus on important information or words by removing the words that have low meaning or low-level information.
- Perform tokenization: this is performed on our dataset to obtain tokens.

This data pre-processing method is based on the Afaan Oromo alphabet and nature. Because those case documents are written in the Afaan Oromo language.

3.4. Handling Imbalanced Data

When the number of data points in one class exceeds the number of data points in other classes, then there exists a class imbalance. This type of problem is common in classifications. The collected dataset for this study contains a somewhat class imbalance that needs to be handled. In this study, the class imbalance problem is solved by synthetic minority oversampling techniques (SMOTE). SMOTE synthesizes new examples as opposed to duplicating examples. In SMOTE, the minority class is over-sampled by taking samples of each minority class and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors (Chawla et al., 2002).

3.5. Feature Extraction

Feature extraction is the method used to translate raw data into a specific input machine-learning algorithm. Before the predictive models can be constructed, the text of the case (lawsuit) must be processed and features must be created from this text. There are various techniques for extracting the feature from the text and converting it into vectors. From those, the following two feature extraction was used in our study (Eklund, 2018).

Bag of words: The BOW model is a pre-processing technique by converting the text into a vector format, which counts the total number of words used in the document most frequently. This model is primarily shown using a table, which contains a word count related to the word itself. In other words, it can be explained as a method to extract features from text documents and use these features for training machine learning algorithms. It tends to create a vocabulary of all the unique words occurring in the training set of the documents. Bag of Words is one of the most fundamental methods to transform tokens into a set of features (Muskan Kothari, 2020). This technique is used to build the vocabulary of all unique words from the corpus. Assume that we have two documents and the bag of word technique has worked as the following.

Document 1: `himatamaan nama ajjeese dhokate`

Document 2: `himatamaan uleedhaan rukutee harkaa cabse`

In the BOW the first step is finding a unique word to build the vocabulary. The unique words from those two documents are {**`himatamaan, nama, ajjeese, dhokate, uleedhaan, rukutee, harkaa, cabse`**}. This is a set of words that are found in the vocabulary. The Bag words technique marks the presence of words as a Boolean value, zero for absent, and one for the present. To visualize the data the BOW uses a table. The following tables show the vector representation of the above two documents.

Table 3.3: BOW feature extraction example

Vocabulary	*himatamaan*	*nama*	*ajjeese*	*dhokate*	*uleedhaan*	*rukutee*	*harkaa*	*cabse*
Document 1	1	1	1	1	0	0	0	0
Document 2	1	0	0	0	1	1	1	1

Finally, we converted the given text to vector as the following:

`himatamaan nama ajjeese dhokate`=[11110000]

`himatamaan uleedhaan rukutee harkaa cabse`=[10001111]

TF-IDF: One method that has proven itself to be easy and effective for features extraction is Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF is an information

retrieval method that can be used to identify the relevance of terms in the documents about a question. (Karniol-tambour, 2013) In this case, it is possible to use attributes to determine which words in a document are most unique to that document.

TF-IDF consists of two stages: first calculating the term TF and then calculating the reverse document frequency (IDF). There are several types of these components. As with count vectors, a TF variant first calculates (*formally defined in equation 3.2*) how often a word occurs in a document.

$$tft, d = \frac{nt,d}{\sum k \ nk,d} \qquad (3.1)$$

Where $n_{t,d}$ is the number of times that term *t* occurs in document *d*, and $n_{k,d}$ is the number of occurrences of every term in document *d*.

To calculate the IDF (*formally defined in equation 3.3*) part of the formula, one alternative is to take the total number of documents in the corpus and divide it by the number of documents where the term appears.

$$idft = log \frac{|D|}{|Dt|} \qquad (3.2)$$

Where $|D|$ is the total number of documents, and $|D_t|$ is the number of documents where the term *t* appears. (Eklund, 2018)

For legal text classification and other textual research, TF-IDF is a better way to convert textual data into a vector space model (VSM). Suppose there is a document with 500 words and out of these 500 words the word "`ajjeese`" appears 40 times, then the Term frequency is will be 40/500 = 0.08, and again suppose there are 100,000 documents and out of these only 1000 documents contains the term "`ajjeese`". Then IDF (`ajjeese`) = 100,000/1000=100, and TF-IDF (`ajjeese`) will be 0.08*100= 8.

3.6. Machine Learning Classification Algorithm

Machine learning algorithms are programs that can learn from data, predict results, and improve performance, without human intervention. The machine learning algorithm can be categorized into supervised, unsupervised, and Reinforcement. The study focuses on the supervised learning algorithm. Supervised learning, making predictions using labeled data. It is widely used for data where there is an accurate map between input and output data (F.Y et al., 2017).

In machine learning, classification; as the name implies classifies data into different classes or predict from which dataset the input data belongs (F.Y et al., 2017). In the classification, there are two popular types of classifications: binary classification, and multi-class

classification. The binary classification is a process of classifies data into two classes. But, multi-class classification is a process of classifies data into more than two classes. In this study, those two classifications were used.

3.6.1. Model Selection Technique

There are many types of supervised machine learning algorithms. We have set some criteria for selecting the best machine learning algorithms for our proposed solution. Because, there is no good concept of how to identify algorithms in problem types, instead, it is recommended that an expert use controlled experiments and find out which algorithm and algorithm configuration works best for a particular classification task (Brownlee, 2020). So, the first criteria are our datasets. Our dataset is collected from OSC and 1638 case documents are prepared and stored in Excel. This data collection is relatively small. So, based on our dataset, the best model that is more powerful on a small dataset was selected. The second criteria are the classification problem. We have a binary and multi-class classification problem. Therefore, we explored or sought out machine learning algorithms that are widely used in both classifications. The third criterion is the machine learning algorithms used to predict judicial decisions in previous research. The fourth criteria are the ability to handle imbalanced data: The dataset collected for this study contains some class imbalance. Some models have a built-in approach to combat class imbalance. In general, we have selected a machine learning algorithm that meets the requirements according to those criteria. The following are some of the algorithms that come under supervised learning that were selected for our study.

Support vector machine: SVM is a supervised algorithm. In this algorithm, the data points are separated into a class with the hyperplane which is as wide as possible from every class. Hyper-plane is constructed using kernel trick which describes whether the problem stated is nonlinear classifier or linear classifier. SVM algorithm gives good results for binary classification with a small dataset. And also it handles multiple classes with extensions. As described in (Adrian Erasmus, 2010), it is one of the most powerful out-of-the-box supervised machine learning algorithms. Unlike many other machine learning algorithms such as neural networks, DNN, and CNN, you don’t have to do a lot of tweaks to obtain good results with SVM.

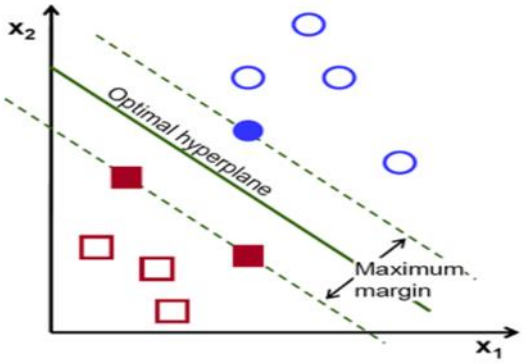

Figure 3.3: SVM with maximized margin.

SVM can be calculated as the following;

$$w * x + b = 0 \qquad (3.3)$$

Where:

w is the support vectors to the hyper-plane,

the intercept (*b*) is found by the learning algorithm,

and *x* is a set input data point.

To predict a new input, the SVM classifier can be written as follows:

$$fw,(x) = g(wx + b) \qquad (3.4)$$

Naïve Bayes: NB is the subset of the Bayes theorem rule, NB was simple to work with a small amount of data and can easily handle multiple classes. KNN and DT are moreover based on the rule, but NB using probability. Using probabilities can sometimes be more effective than using hard rules for classification and you can reduce the need for a lot of data by assuming conditional independence among the features in your data (Tutorial Points, 2020a).

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \qquad (3.5)$$

Random Forest: Random forest is a supervised learning algorithm. Forest is an ensemble of decision trees, often trained with bagging techniques. Random forest builds many decision trees and combines them to get a more accurate and stable forecast. Due to this, the random forest classifications facilitate the reduction of the model's over-fitting (Tutorial Points, 2020b).

3.6.2. Hyper-parameter Tuning

Tuning or optimization the hyper-parameter (Agrawal, 2021) is a method of selecting the best set of optimal hyper-parameters for a learning algorithm. The same machine learning model may require different limits, weights, or learning levels to generalize different data patterns. These measures are called hyper-parameters, and the model needs to be tuned to optimally solve the machine learning problem. To produce an optimal model, hyper-parameter techniques find a tuple of hyper-parameters that reduces a predefined loss function on given independent data. There are two major ways of searching parameters for an optimal model. It is called grid search and random search.

Grid search is an effective exhaustive search for every value that can be chosen manually. The grid search algorithm must be guided by certain performance metrics, especially measured on the training set by cross-validation. But, grid search is a very popular method for optimizing the hyper-parameters in practice. Random search can replace the traditional grid search by randomly selecting.

3.7. PJD Model Evaluation

3.7.1. Prediction Model Evaluation

The prediction model needs to be evaluated by model evaluation techniques to ensure that the model is compatible with the dataset and works well on new unseen input data. The purpose of model performance evaluation is to estimate the overall accuracy of a model based on unseen/ external sample data (Raschka, 2018). The performance evaluation method is divided into two categories; holdout and Cross-validation (Singh, 2019).

a) Holdout

In this method, the largest data set is randomly divided into three sub-categories:

- The training set is a subset of the dataset to build predictive models.
- The validation set is a subset of the dataset to evaluate model performance developed at the training level. Provides a test platform to fine-tune the model parameters and select the best model. Not all modeling algorithms require a validation set.
- Test sets are a subset of a dataset to evaluate the future performance of a model. If a model fits the training set much better than it fits the test set, over-fitting is probably the cause.

On the other hand, the holdout is dependent on only one train-test split. That makes the hold-out technique score dependent on how the dataset is divided into train and test sets. (Allibhai, 2018) When there is a large dataset, the holdout method is better.

b) Cross-validation

Cross-validation is one of the performance evaluation methods for evaluating and comparing models by dividing data into partitions: one is used to train a model and the rest is used to test or validate the model. To avoid over-fitting, the cross-validation technique is the most widely used evaluation method (Stone, 1974). Different types of cross-validation can be used as an evaluation method.

K-fold cross-validation: in the k-fold, the first samples are randomly divided into k equal-sized sub-samples. Of the k sub-samples, one subsample remains as the validation data set for testing the model, and the remaining k − 1 sub-sample serves as the training dataset (Brownlee, 2018). The average of k scored accuracy is called CV accuracy and served as a performance metric of the model.

Stratified K-Fold Cross Validation: Because K-fold cv randomly modifies the data set and divides it into folds, the chances of getting very unbalanced folds are high, which makes model training to be biased. The stratified form of k-fold CV, which conserves the imbalanced class distribution in each fold is called stratified k-fold CV, and It forces class distribution in each split or divide of the data to match the distribution in a complete training set (Hussain Mujtaba, 2020). In this study stratified k-fold cross-validation technique was used because the data collected contains imbalanced class distribution.

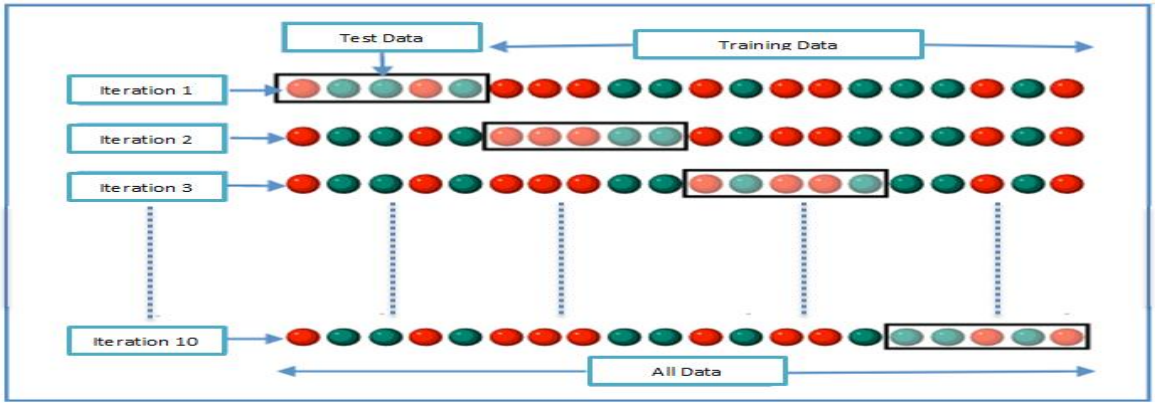

Figure 3.4: 10-fold cross-validation [15]

[15] https://upload.wikimedia.org/wikipedia/commons/b/b5/K-fold_cross_validation_EN.svg

3.7.2. Classification Metrics

Since classification algorithms do not produce the same results, predictive classification model evaluation metrics are needed to measure model performance. There is a different kind of classification model evaluation (M & M.N, 2015). Of these, precision, recall, confusion matrix, and f1-score are used in this study with cross-validation. The followings are the fundamental terms in performance measure:

- *True positive (TP)*: is the condition when both actual value and predicted value are true.
- *True negative (TN)*: is the condition when both the actual value of the data point and the predicted are False.
- *False-positive (FP)*: These are the cases when the actual value of the data point was False and the predicted is true.
- *False-negative (FN)*: are the cases when the actual value of the data point was true and the predicted is False.

a) Confusion Matrix

The confusion matrix in Table [3.1] is used to explain how these parameters are calculated. Here, the values for false negative (FN) and false positive (FP) give, for a given lawsuit (sue), the number of Judgments that are incorrectly predicted as guilty (yakkama) and not guilty (Bilisa) respectfully. Similarly, the values for true negative (TN) and true positive (TP) give the number of Judgments correctly predicted guilty (yakkama) and not guilty (Bilisa) respectfully.

Table 3.4: Confusion matrix with binary classification

		Prediction	
		Guilty (Yakkama)	not guilty (Bilisa)
Actual	Guilty (Yakkama)	True Negative (TN)	False Positive (FP)
	not guilty (Bilisa)	False Negative (FN)	True positive (TP)

Table 3.5: Confusion matrix with multi-class classification

		Prediction				
		Category 1	Category 2	Category 3	Category 4	…….. Category 38
Actual	Category 1	True Category 1	False Category 2	False Category 3	False Category 4	False Category 38
	Category 2	False Category 1	True Category 2	False Category 3	False Category 4	False Category 38
	Category 3	False Category 1	False Category 2	True Category 3	False Category 4	False Category 38
	Category 4	False Category 1	False Category 2	False Category 3	True Category 4	False Category 38
	……. Category 38	False Category 1	False Category 2	False Category 3	False Category 4	True Category 38

b) Accuracy

Accuracy is calculated for each lawsuit (sue) and the models. Equation 3.6 gives the formula for the accuracy metric. For a given lawsuit (sue), this gives the proportion of Judgment outcomes that were correctly predicted.

$$\text{Accuracy} = \frac{TN+TP}{TN+FP+FN+TP} \qquad (3.6)$$

c) Precision and Recall

Precision is the ratio between the True Positives and all the Positives. And also recall, tells us how much of the actual positive cases we were able to predict correctly with our model. For a given case, precision gives the proportion of all guilt's predictions that were correct. Recall, on the other hand, gives the proportion of actual guilt that was correctly predicted. The formula of precision and recall are given by Equations 3.7 and 3.8 respectfully (Purva Huilgol, 2020).

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP)+False\ Positive\ (FP)} \qquad (3.7)$$

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP)+False\ Negative\ (FF)} \quad (3.8)$$

d) F1-Score

The F1-score is a harmonic mean of Precision and Recall, so it offers a combination idea of these two metrics. It is highest when Precision is equal to Recall. Mathematically:

$$\mathrm{F1} = 2\ x\ \frac{precision\ x\ recall}{precision+recall} \quad (3.6)$$

3.8. Research Tools

To develop the proposed model the following different tools were used.

3.8.1. Design Tools

It is important to choose the best design tool to describe and demonstrate our proposed system in detail. Some of the most popular and well-known design tools software, such as Edraw Max and Draw.io, has been selected for the design of the prototype and the architecture of the proposed model.

Edraw Max is UML builder software that helps to create illustrations using ready-made symbols and templates as explained in [16]. It allows you to import drawing into different file formats such as Word, HTML, PDF, PPT, etc. In addition, it provides a user-friendly interface similar to MS Word and allows you to share designs anytime and anywhere. In particular, we used it to draw organizational structure, proposed system architecture, and research methodology workflow.

Draw.IO [17] is a free online UML tool. It is one of the best UML tools that allow users to easily create and manage a drawing. Depending on your needs, Draw.io has options for storing saved charts in the cloud, on other a server, or in other a data center. To drawing a prototype these tools were used.

3.8.2. Data Preparation Tools

The data collected is not editable text; it is composed as a PDF but is a scanned document. Software that converts a PDF file into an editable text is needed to extract important text such as lawsuits (fact), decisions, and other text. For this purpose, MS-office OneNote was selected. Finally, the data which changed to editable was stored in Ms-Excel.

[16] https://www.edrawsoft.com/
[17] https://www.guru99.com/best-uml-tools.html

MS-office OneNote: OneNote is a part of Microsoft Office, along with programs like Word, Excel, publisher, and PowerPoint. As described in (Johnson, 2020), it's a popular application software designed for research, note-taking, and information storage. In addition, (Microsoft, 2021) OneNote supports Optical Character Recognition (OCR), a tool that allows you to copy text from a file printer or picture and paste it into your notes to make changes to words. OneNote is a great way to do things, such as copying information or text from the case document scanned. Of all the documents, a specific area containing lawsuits or charges has been converted to editorial text through OneNote.

MS-Excel: Spreadsheet applications like MS Excel use a set of cells arranged into columns and rows to manipulate and organize data. MS Excel allows users to arrange data to view different factors from various perceptions. It's used to storing cleaned and filtered selected data in the form of tabular and also it was used to manage the prepared dataset (Techopedia, 2020).

3.8.3. Implementation Tools

Choosing the right tool can be as important as working with the best algorithms in machine learning. Therefore, the following tools have been used in the implementation of the proposed model.

a) Programming Tools

There is some very popular machine learning languages in machine learning, such as Python, R, and so on. The Python program language was used to develop our proposed models. Python is a high-level programming, object-oriented, general-purpose interpreted, and interactive, and also it's a popular language with high-quality machine learning and data analysis libraries (Tutorial Points, 2017).

b) Data analytics and visualization tools

Pandas: a Python data analysis library enhancing analytics and modeling. Pandas simplify the analysis by converting JSON, CSV, and the SQL database or TSV data files into a data frame, Excel, or SPSS tables with rows and columns [[18]].

Matplotlib is a Python machine learning library for quality visualizations. Matplotlib is a Python two-dimensional (2D) plotting library. Matplotlib allows you to generate

[18] https://pandas.pydata.org/

production-quality visualizations with few code lines. Plotting is a visualization of ML data [19].

Jupyter notebook: collaborative work capabilities. The Jupyter Notebook is a free web application for interactive computing. Notebook is rich in functionality and offers a variety of usage options [20]. We used this application to write, run a python program and for data processing and visualization while we developed the proposed model.

c) Frameworks

NumPy: an extension package for scientific computing with Python. It supports multidimensional arrays and matrices. In ML data can be represented in form of arrays [21]. NumPy quickly and easily integrates with many kinds of databases. We used this library while we converted the data into the vector and reshaping the vector to fit the vector with the proposed model.

Scikit-learn [22]: easy-to-use machine learning framework for numerous industries. Scikit-learn is open-source. Python machine learning library built on top of SciPy (Scientific Python), NumPy, and matplotlib. We used this library in the proposed model development for splitting the dataset into train data, validation data, and test data.

NLTK: Python-based human language data processing platform. NLTK is a platform for the development of Python programs to work with human language. We used this library for applying data preprocessing techniques that applied to the development of the proposed model.

3.8.4. Deployment Tools

Flask [23]: After the experiments are done the best performing model is deployed on a web server, where it can be used to make a real-time prediction by providing the story needed to be verified. To realize this, we used a flask micro web framework. Because we are deployed in machine learning.

3.8.5. Hardware Tools

The tools which are discussed above in this section have been deployed on a personal computer equipped with a Processor Intel(R) Core(TM) i3-2350M CPU @ 2.30GHz, 2300

[19] https://matplotlib.org/
[20] https://jupyter.org/
[21] https://www.numpy.org/
[22] https://scikit-learn.org/stable/index.html
[23] https://palletsprojects.com/p/flask/

Mhz, 2 Core(s), 4 Logical Processor(s), 4 Gigabyte of physical memory, 465 Gigabyte hard disk storage capacities. The operating system is Windows 10 Pro, 64 bits.

3.9. User Interface Design

We designed a user interface using a python web framework called Flask. To design the user interface we used Bootstrap that is an open-source framework for designing a responsive web application. The goal of designing a user interface is to test our model using new and unseen data that is inputted by an expert.

Summary

Our objective is to predict the judicial decision of the Oromia Supreme Court using a machine learning approach. A variety of research methods, tools, and techniques have been used to achieve this objective. We used a quantitative research method for our study because the study answers the research question through the experiment. It is necessary or mandatory to build a dataset for experiments. As a result, the PJD dataset has been build using the case documents that are collected from OSC. Various NLP techniques have been used to clean up and make suitable data for the machine learning model. Naïve Bayes, the support vector machine, the random forest was selected for model building, with TF-IDF and Bag of word feature extraction. The evaluation method, like cross-validation, confusion matrix, accuracy, precision, recall, and f1 score was used to evaluate the model. To design model architecture, user interface, building dataset, and to write implementation code different tools, packages, and libraries were used.

CHAPTER FOUR

4. PROPOSED SOLUTION FOR JUDICIAL DECISION PREDICTION

4.1. Introduction

In this chapter, the design and overview of the proposed solution have been discussed. The proposed model architecture shows how a judicial decision prediction model would be designed and developed in order to solve the problems and answer the questions raised in chapter one, following to fill the gaps mentioned in the literature review. The proposed solution contains the way of preparing datasets and different types of ML algorithms and NLP techniques that are used to build the PJD model and model evaluation techniques that are used to evaluate models.

4.2. Proposed Model Architecture

Model architecture is used to show how a judicial decision prediction model or proposed model is developed. In this, two prediction models were constructed. Those are the judgment prediction model and penalty prediction model. The proposed judgment prediction model architecture is used to predict whether the defendant is guilty or not guilty. And, the proposed penalty prediction model architecture is used to predict the year of imprisonment based on the defendant's commit crime. Therefore, to achieve these tasks, we divided our process into different phases.

The first phase is about data. In this phase, we prepared the data which is collected from the OSC and applied the data preprocessing technique to convert the sentences into tokens to make them more suitable for the ML algorithm as discussed in chapter three, section 3.3.

The second phase is about feature extraction. Feature extraction is important to convert raw data into computer understandable form. In this study, we propose two feature extractions, TFIDF and Bag of word to convert tokens to vectors. After we have been converted tokens to vectors, SVM, NB, and RF ML algorithms would be applied to learn patterns from the vectors. Those algorithms were selected according to the criteria set out in Chapter three in section 3.5.1. So, in this study, SVM, NB, and RF ML algorithms are used to build a predictive model. Those models were evaluated using stratified 10 fold cross-validation

techniques and classification metrics. The best model would be selected, based on the accuracy they are scored.

The final selected judicial decision model was used to develop a prototype that can take new sue texts as input and predict the input text judgment and penalty. Proposed model architecture is provided to clarify the flow of research as follows.

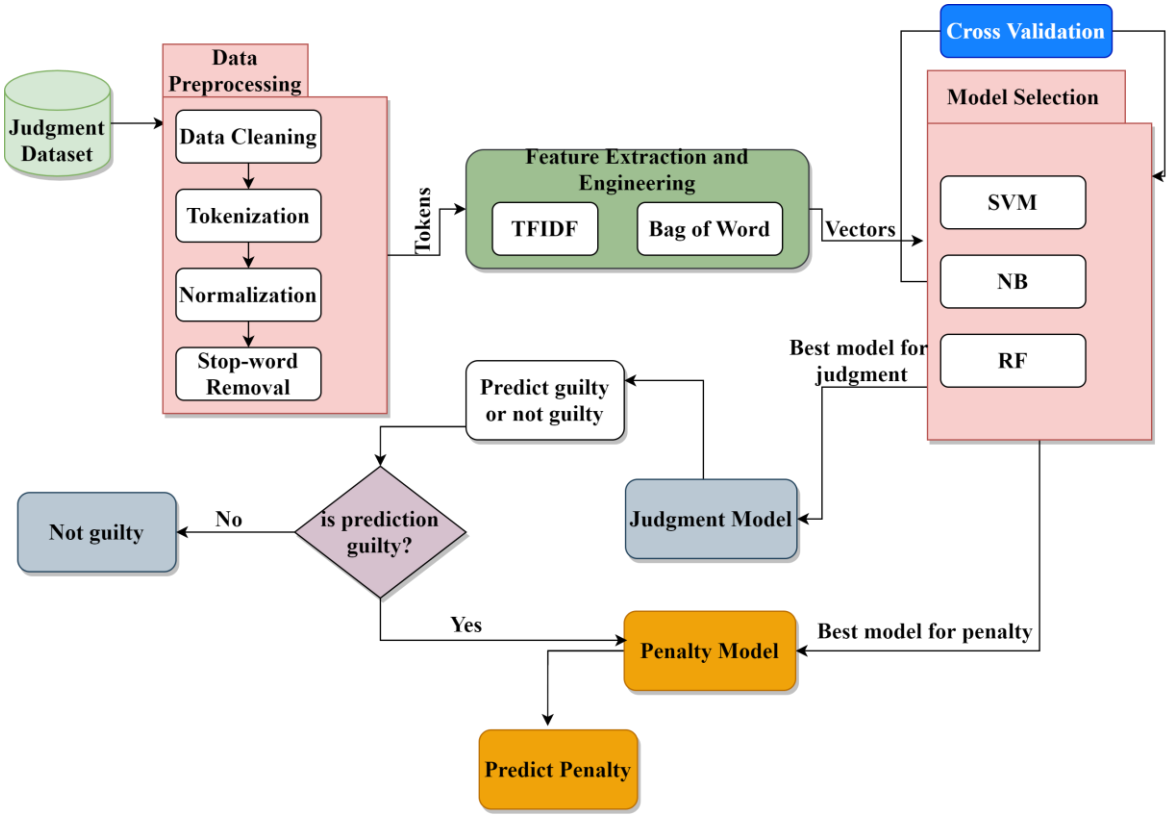

Figure 4.1: Judicial decision prediction model architecture

4.2.1. Judgment Dataset

This study aims to develop a prediction judicial decision using machine-learning techniques. Therefore, to build the model, a new PJD dataset must be built. This new dataset is needed because there is currently no published or prepared dataset for judicial decision prediction in OSC, not only in OSC but also in Ethiopia. The process of building the dataset for the PJD involves the main steps of collecting data from the OSC registry and preparing the dataset in a suitable way for model building.

Therefore, we have built a PJD dataset using case documents collected from the OSC that was written by the Afaan Oromo language. Due to a lack of complete information and time, not all case type was used. Only the criminal case that related to murder and body injury have been collected and constructed.

4.2.2. **Proposed Data Preprocessing**

One of the critical problems of training machine-learning models is the data itself. The collected data for this study contains some inconsistent data that need to be cleaned and normalized. So, this phase performs preprocessing of lawsuits or cases prepared by Afaan Oromo for the training and testing prediction judicial decision model. In this phase, we have been discussed details of how the data preprocessing would be done.

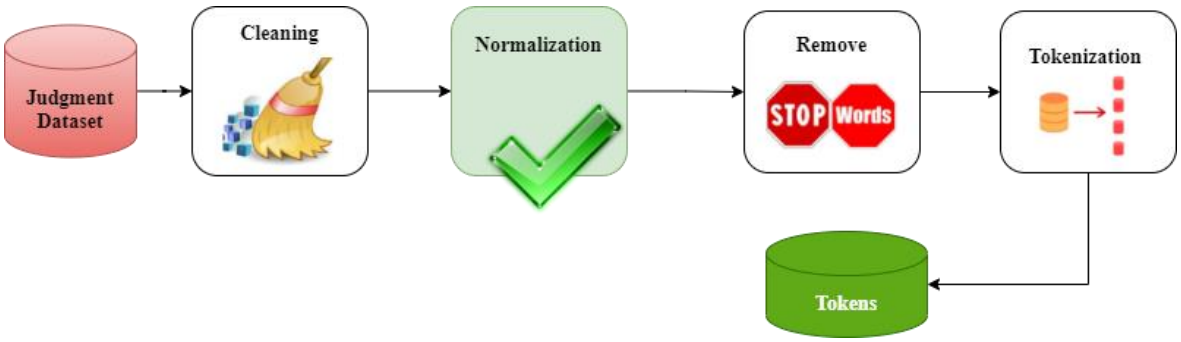

Figure 4.2: Proposed data preprocessing technique

*4.2.2.1.**Cleaning***

To avoid unnecessary content and use it, we go through some cleaning steps to make the data more representative. To cleaning the prepared dataset, two methods are used. To removes the name of the defendant, the plaintiff, the place, or the area, and unnecessary words such as `'Gaafa, Guyya, jedhaamu'`, the manual method was used. And the second is the NLP technique (method); this cleanser removes all special characters, symbols, punctuations, extra space, and others that are not needed and converting all text into lower case.

Algorithm 3.1: Cleaning method [24]

OBTAIN: unprocessed dataset

OUTPUT: clean dataset

INIT:

1: Read the dataset

2: WHILE (it is not the end of file):

[24] `https://towardsdatascience.com/nlp-in-python-data-cleaning`

IF text contain special characters and stmbols [',', '.', "", ':', ')', '(', '-', '!', '?', '|', ';', """, '$', '&', '/', '[', ']', '>', '%', '=', '#', '*', '+', '\\', '•', '~', '@', '£', '•', '_', '{', '}']

THEN

Eliminate characters

IF text contain [tabs, extra white space] THEN

Eliminate

IF text contain [a-z A-Z] [0-9] THEN

Eliminate Ethiopic numbers

Return processed text

ENDIF

3: HALT

---

*4.2.2.2.**Tokenization***

After some preprocessing techniques clean the dataset, features in the dataset should take vectors form. Here the next process is applied, tokenization. The process of tokenization divides the text data into pieces (or tokens), and often also removes certain special characters such as apostrophes, commas, and periods (Allahyari et al., 2017). The text is also often normalized to be lower-case only. The tokens usually consist of either a single word or what is called an N-gram, meaning that N consecutive words are split into a single token (Abinash Tripathy and Ankit Agrawal and Santanu Kumar Rath, 2016). The idea is to preserve some of the information that is stored in the order of the words. For instance, the sentence "`himatamaan uleedhaan rukutee ajjeese` (Defendant struck him with a stick and killed him)" would be split into the following tokens when using 1-grams: `“himatamaan”, “uleedhaan”,”rukutee “, and “ajjeese”`.

*4.2.2.3.**Normalization***

Normalization is one of the steps used to get clean data from unstructured text. There are two types of normalization; character-level normalization and word-level normalization. Character-level normalization is different characters that have the same sound but are written in different forms. Whereas word-level normalization is written in different forms but they have the same meaning. For instance, words like `“dhakaa” and “dhagaa”, “ishii” and “ishee”` have the same meaning. In this study, word-level normalization is used. The remaining normalized words are listed in Appendix A.

Algorithm 3.2: Normalizing method [25]

OBTAIN: unnormalized dataset

OUTPUT: clean dataset

INIT:

1: Read the dataset

2: WHILE (it is not the end of file):

IF data contains [yennaa] THEN

replace with 'yeroo'

IF data contains [dhakaa] THEN

replace with 'dhagaa'…..

Return normalized text

ENDIF

3: HALT

*4.2.2.4.**Remove Stop-words***

Stop words are common words that are filtered out before vectoring. After completing text cleaning and tokenization, we implemented stop-word removal to make our dataset more suitable for machine learning algorithms. Afaan Oromoo stop-words like [`kanaaf, irraa,` and others] were collected from a different source before being done (Tesfaye, 2010). The left Afaan Oromoo stop-words are listed in Appendix B.

Algorithm 3.3: Stop-words Removal [26]

OBTAIN: Tokens dataset

OUTPUT: Tokens dataset free from stop-words

INIT:

1: Read the dataset

2: WHILE (Tokens dataset):

IF tokens dataset not in ListOfStopwords: THEN

Append the word in ListOfRemovedStopWord

Return ListOfRemovedStopWord

ENDIF

3: HALT

[25] `https://towardsdatascience.com/text-normalization`

[26] `https://www.geeksforgeeks.org/removing-stop-words`

4.2.3. Feature Extraction

When learning machines using machine learning algorithms and working with textual data, machine learning algorithms cannot be applied directly to the raw text. Therefore, we convert text data to vectors using feature extraction techniques. The study proposed two features extraction, the bag of the word, and TFIDF, to convert the data into vectors. These methods take token data sets and convert them to vectors using the feature extractions as shown in Figure 4.3.

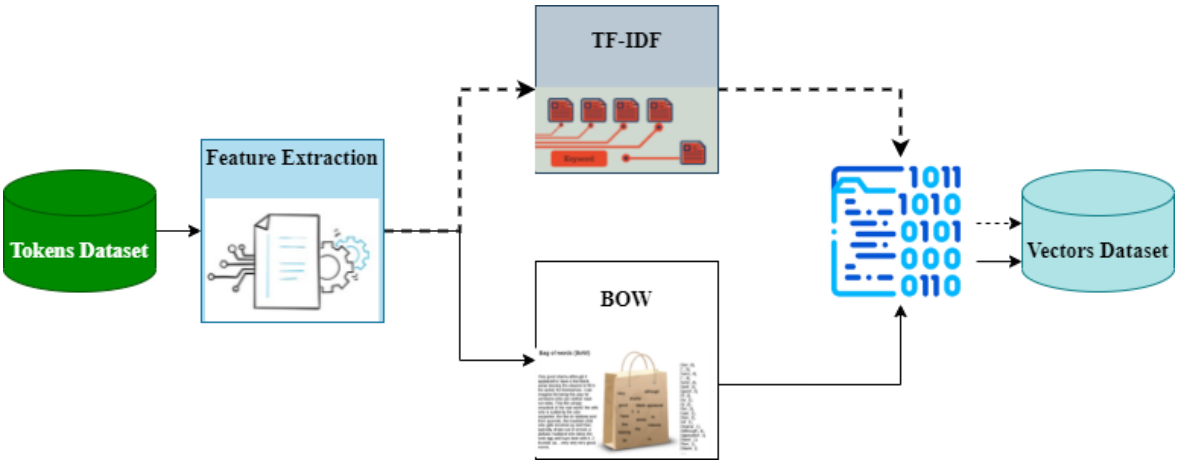

Figure 4.3: BOW and TF-IDF feature extraction

As described in chapter one section 3.4, the bag of the word creates a set of vectors containing the number of word events in the document, whereas, the TF-IDF model contains information on the most important ones and the most important words. To make a comparison between them we proposed those two feature extraction.

4.2.4. Proposed Machine Learning Classification Algorithm

The objective of this study is to predict judicial decisions using a machine learning algorithm on categorized and labeled datasets. In this study, we proposed binary class classification and multi-class classification. That is, we have categorized our dataset to predict judgment and punishment. This is because the outcome of a judgment prediction serves as an input or another feature for punishment prediction and also there are other features that determine the punishment. Predict judgment is binary classification; it has two classes guilty and not guilty. Whereas, predict punishment is a multi-class classification; it has thirty-eight classes. The proposed model first classifies the defendant's guilt or not guilty. If the defendant is guilty, he or she will be subject to a penalty related to the crime.

However, as shown in figure 4.4, the architecture of both judgment and penalty models has been drawn as one model to reduce the complexity of architecture.

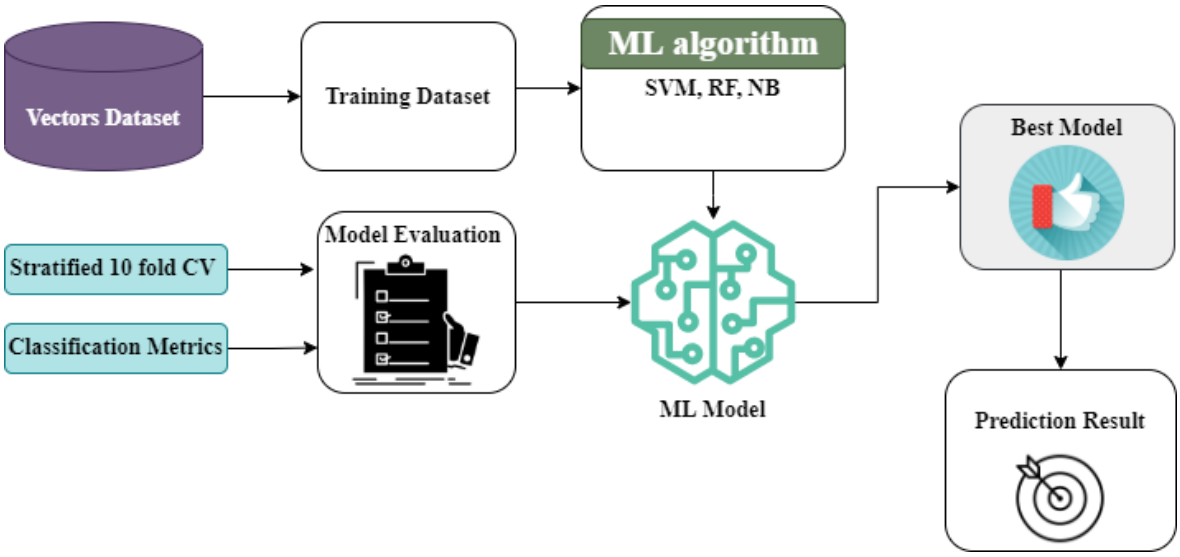

Figure 4.4: Model training diagram

In this study, three machine learning algorithms would be proposed for both classification problems. These are as follows:

Support vector machine*:* SVM can handle both classifications. There are several hyper-planes that can be selected in the SVM to identify the two data points. In SVM, our goal is to find hyper-planes with the maximum margins. Basic SVM only supports binary classification, but extensions are also proposed to handle multiclass classification. To classify multiple groups of classes, we used a version of the SVM algorithm called the One-vs.-rest (OVR) method for multi-class classification. The OVR method identifies each class from the rest of the classes which is in the data set.

How Support vector machine works: Support Vector Machine provides a hyperplane line that takes into account all data points and divides both parts. This line is called the "Decision boundary". The SVM algorithm finds the closest point in both sections. These points are called support vectors. The distance between vectors and planes is called the margin. And SVM's goal is to increase this margin. A High-margin hyperplane is called an optimal hyperplane (Dwivedi, 2020).

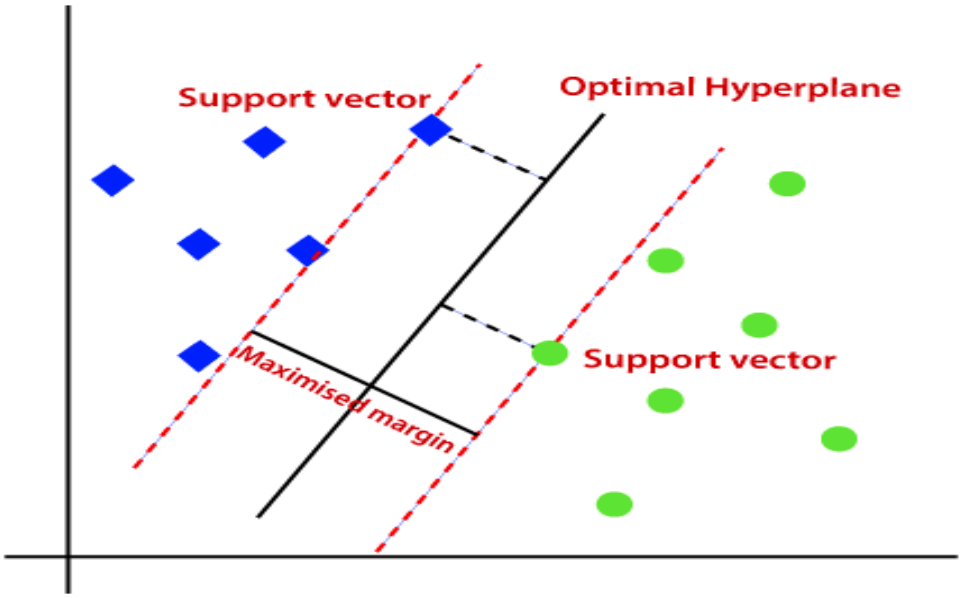

Figure 4.5: Support vector machine algorithm

Random Forest*:* Random forests are an ensemble learning method for classification, they give good results even without the use of hyper-parameters. As the name indicates this algorithm randomly creates a forest with several decision trees. In this study, a random forest algorithm was used to build judicial decision classification models (Tutorial Points, 2020b).

How Random Forest Works: RF makes the final output predictions based on the majority votes that are taken from each tree rather than relying on a single decision tree. The number of trees in the forest leads to good accuracy and prevents overfitting. As depicted in Figure 4.8, a random forest algorithm is made up of many decision trees to predict the correct result. Those different decision trees were trained with a prepared training dataset, and then they predict the outcome. Finally, it selects the highest predictive score using the majority vote and uses this result as the final result (Tutorial Points, 2020b).

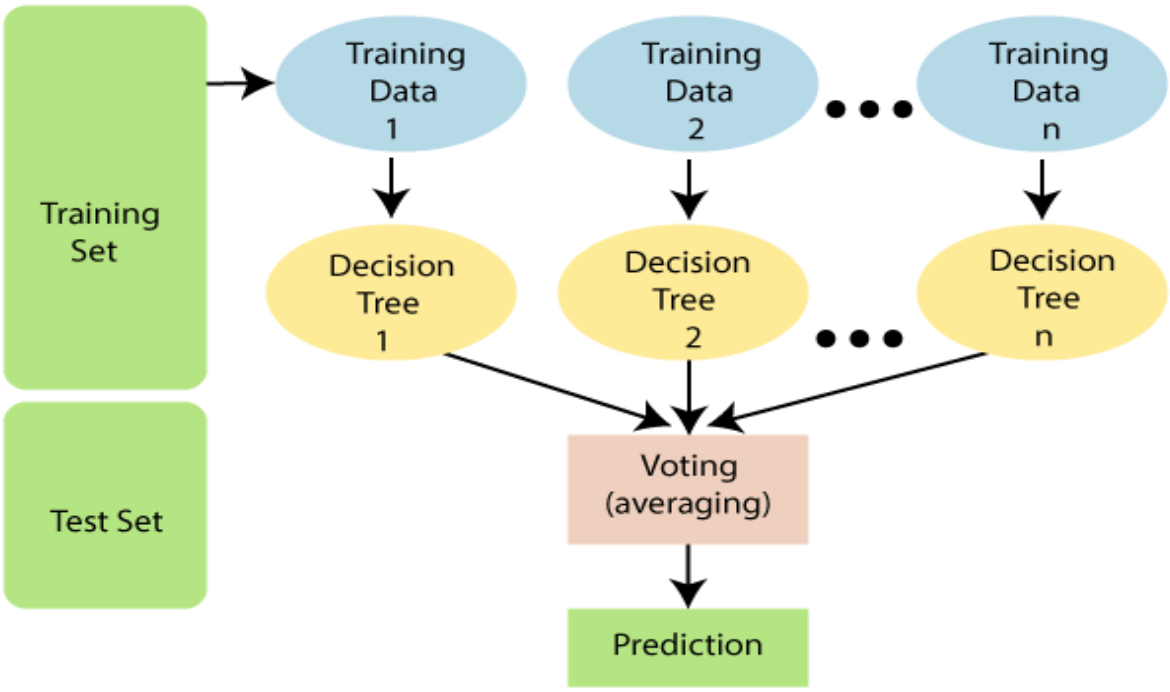

Figure 4.6: The working of the RF algorithm

Random forests operate in two stages: the first is to combine the N decision tree to create a random forest, and the second is to make predictions for each tree in the first step.

Naïve Bayes: it is a kind of classifier that uses a Bayes theorem and it works on conditional probability. Bayesian probability and Baye's rule give us a way to estimate unknown probabilities from known values. NB predicts membership opportunities for each class, such as the probability that a given record or data point belongs to a specific class. It works on the Bayes theory of probability to predict the class of unknown data sets (Tutorial Points, 2020a).

How Naïve Bayes Works: NB algorithm mainly focused on conditional probability. We can calculate the probability of an event using its prior knowledge in conditional probability. The following steps depict how the NB algorithm would be working (Tutorial Points, 2020a).

Step 1: dataset converted into frequency tables.

Step 2: finding the probability of a given feature to generate the likelihood

Step 3: calculate the posterior probability using the Bayes theorem

4.2.4.1. **Hyper-parameter Tuning**

It is the method of choosing a set of hyper-parameters for a machine learning algorithm that allows the model to optimize its performance. Hyper-parameter setting in the right way is the only method to extract the highest performance out of the models. This can be done in either manual or automatic based hyper-parameter setting. In the former method, different groups of hyper-parameters are set and experimented on a manual basis. This is tedious, and may not be applicable in cases with multiple hyper-parameters to attempt (Liashchynskyi & Liashchynskyi, 2019). So, the best hyper-parameters are found using an algorithm that automates and optimizes the process. The second method is used in this study. Grid search and random search optimization for hyper-parameters are the two most commonly used algorithms.

4.2.5. **Proposed Model Evaluation**

To evaluate our model's different evaluation techniques were used in our study. As we discussed in chapter 3 section 3.6, cross-validation, accuracy, precision, recall, confusion matrix, and f1-score are proposed to evaluate our model. Additionally, our model was evaluated by six law experts as human evaluation.

Cross-Validation: In the CV approach, the whole set of data would be divided into k equal-sized groups or folds. And each fold is considered a validation set, the rest of the k-1 serves as a training set to fit the model.

Algorithm 3.4: Cross-Validation [27]

OBTAIN: Training Set

OUTPUT: k-folds split data

INIT:

1: Read the dataset

2: For each group:

Take the group test data set

Take the left groups as a training data set

Fit the model

Hold the evaluation score and drop the model

3: Return summary of evaluation scores

4.3. Proposed Model Prototype Architecture

At this phase, we tried to design the prototype for the proposed system like as in the following figure. A different tool was used to develop the prototype of the proposed model as we discussed in the methodology phase. Once the models have been trained and validated on each sample of cross-validation, the best-performing classifier is deployed on the webserver. From the following architecture, the user sends the request via HTTP. Then, the flask server receives an HTTP request from the user and submits the request to the model. Finally, the model responds to the user's request based on their request.

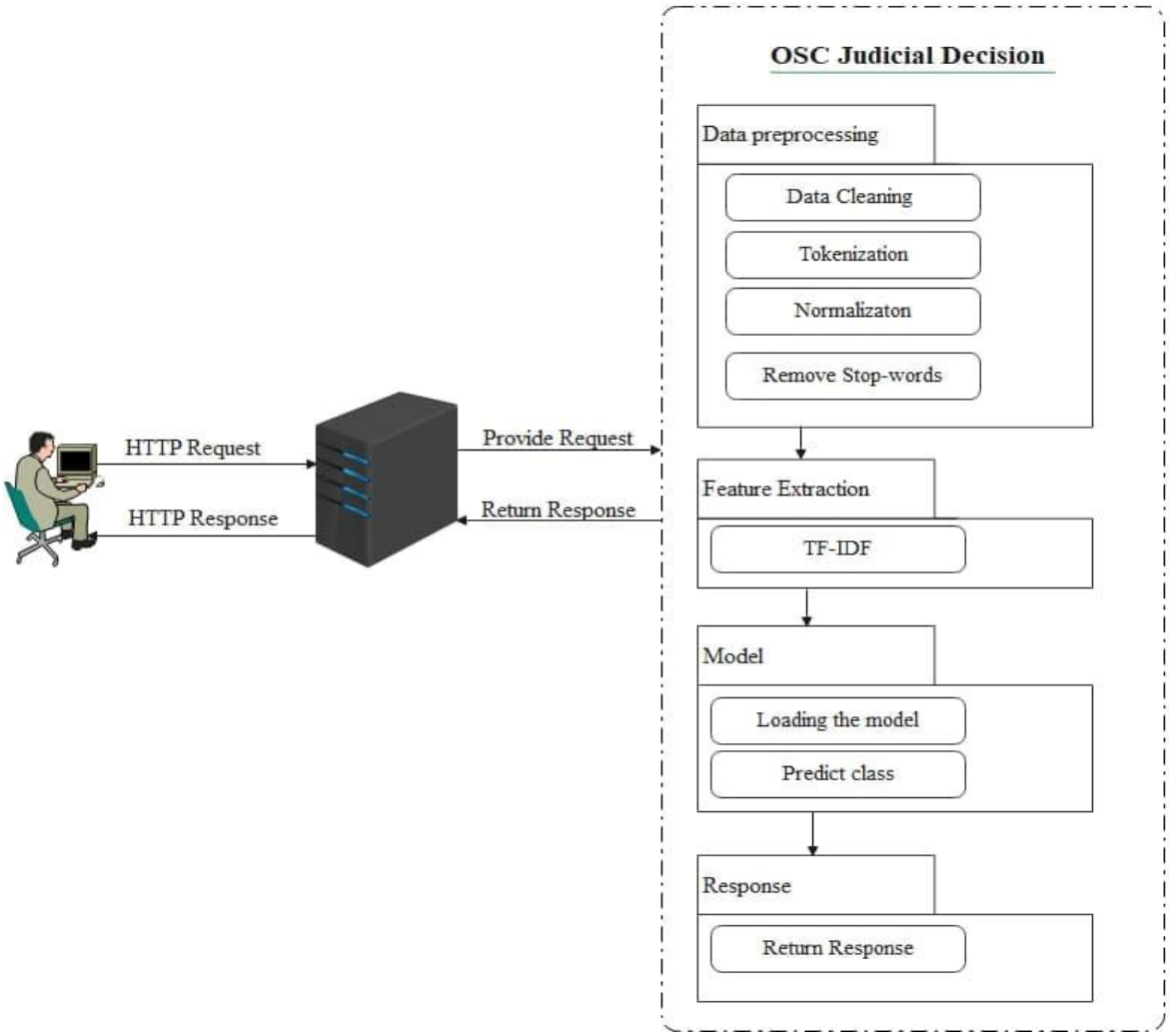

Figure 4.7: Prototype of judicial decision

[27] https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right

CHAPTER FIVE

5. EXPERIMENTATION AND IMPLEMENTATION

5.1. Introduction

In this chapter, the implementation and experimentation of the proposed solution would be discussed detail. Various experiments have been made to solve the problem and answer the research questions raised in chapters one, sections 1.3 and 1.4. The first group of experiments aimed to identify the way of preparing data. Then, the dataset prepared would be loaded into python programming and ready for experiments. By using python code, we have been making suitable data for machine learning models by applying different data preprocessing techniques. The end group of experiments aims to train the selected machine learning model with selected feature extraction and evaluate their performance using model performance evaluation. The finally best-performed model would be integrated into the flask server. Generally, in the section, the implementation of all tasks would be discussed detail.

5.2. Implementation Environment

In this study, we used a variety of development tools and packages to implement the proposed solution. As we discussed in chapter three section 3.7, we were used Python programming language to develop our model. Pandas and mat-plot-lib were used for data analysis and visualization. And also NLTK libraries have been used for data preprocessing. In addition to this, many tools and libraries were used to implement the proposed judicial decision model.

5.3. Dataset Description

We built the dataset that contains judicial decisions (judgment and punishments) written in the Afaan Oromo language. This dataset is made up of nine independent and two dependent attributes. The features or attributes included in this dataset are articles, charges, confession, and witness of the prosecutor, the witness of the defendant, judgment, and idea of mitigating punishment, the idea of increasing the punishment, and penalty or punishment. For our experiments, we extracted and reorganized these feature labels as datasets. We have used 1638 criminal case documents that were judged by the Oromia Supreme Court for these experiments. As we discussed in the preceding chapter, not all

criminal cases are selected. We have only used cases related to murder and body injury. The following table depicted the type of features with their description.

Table 5.1: Features description

No	***Features***	***The unique name of the features***	***Description***
1	Law Article	Kewwata	It is a part of the Constitution that is assigned to prosecute a criminal based on committed crimes.
2	Charge	himata	It contains a statement of the defendant's committed crimes against the plaintiff
3	Confession	wakkatera	It contains the defendant's confession that the defendant admitted or did not believe in the alleged crime.
3	The witness of the prosecutor	raggasisera	Written or human evidence proving the crime committed.
4	Defense witnesses	Ittisa_raga	Written or human evidence that is defense from the accused.
5	Judgment	Murte	This section contains the decision of whether the defendant is guilty or not guilty
6	The idea of mitigating punishment	YA_salphisu	The accused will raise the factors according to Art 82/1/A that mitigating a punishment as the possibility to hear the case and decide is nearer.
7	The idea of increasing punishment	YA_cimsu	The prosecutor will raise the factors that increasing punishment on the defendant if the committed crime is difficult.
8	punishment	Gulanta	is the stage of punishment with the beginning

	stage		and end of punishment
9	Punishment	adabbii	After the court decided the accused person is guilty, the punishment would be implemented

5.4. Importing Libraries

Python programming languages have many and varied libraries. The library is a set of built-in Python modules used to perform various tasks. To perform different experiments, different libraries were imported that are depicted in the below figure.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC, SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_val_predict, KFold
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn import preprocessing
import re
import nltk
from nltk.corpus import stopwords
import string
from nltk.tokenize import word_tokenize
from sklearn.linear_model import LogisticRegression
```

Figure 5.1: Sample code of importing libraries

5.5. Analyzing Data

In machine learning, we must use datasets for data analytics and experimentation when running Python programs. Python has a variety of modules that allow importing external data into a Python program in a variety of file formats. The prepared dataset is stored in Excel and has been saved with CSV file extensions. So, as you can see in the figure 5.2, we have loaded our dataset to the Python program using the following code. To encoding character, cp1252 was used. Cp1252 is especially used to encode Latin alphabets.

```
df = pd.read_csv("bincyali.csv" ,  encoding="cp1252")
```

Figure 5.2: Sample code of loading dataset

5.6. Data Preprocessing Implementation

5.6.1. Implementation of Data Cleaning

After the dataset was loaded onto a python program, we performed data preprocessing to make suitable our dataset to machine learning algorithms. To perform data preprocessing python RegEx (regular expression) re and NLTK modules were used. Using those modules, we first remove and replace punctuation marks, special characters, symbols, and unwanted characters from our datasets. The method then returns the cleaned datasets.

```
#remove white space
df["case"] = [string.strip() for string in df["case"]]
#Remove periods
df["case"] = [string.replace(".", "") for string in df["case"]]
```

```
#convert to lower case
def LowerCase (string: str) -> str:
    return string.lower()

#Apply function
df["case"]=[LowerCase (string) for string in df["case"]]
```

```
#Remove punctuation
#Load libraries
import unicodedata
import sys
#Create a dictionary of punctuation characters
punctuation = dict.fromkeys(i for i in range(sys.maxunicode)
                            if unicodedata.category(chr(i)).startswith('P'))

#For each string, remove any punctuation characters
df["case"]=[string.translate(punctuation) for string in df["case"]]
```

Figure 5.3: Implementation of cleaning dataset

5.6.2. Implementation of Normalization

Normalization of words was implemented to normalize the text using RegEx. The following figure 5.4 shows the normalization code.

```
# Dictionary of words to be norimalized
word_dic = { " otuu": " osoo "," otoo": " osoo "," fundura ": " fulduraa "," fuula ": " fula "," ishii ": " ishee ",
                    " of-eeggannoo":" of eeggannoo "," dhakaa ": "dhagaa "," dhaka ": " dhagaa"," diree ": " waraane ",
                    " gayee ": " gahee "," hiija": " haalo "," hiijaa ": " haalo "," yenna ": " yeroo "," tara ": " yeroo "
                    " kessaa ": " keessaa "," marsaa ": " yeroo "," dhaqabsiiseef ": " geessiseen ",
                    " himatamtertii ": " himatamtee jirti "," himatamteertii ": " himatamtee jirti ",
                    " himatamera": " himatamee jira "," himatameera": "himatamee jira "," iddoo": " bakka ",
                    " tan": " kan "," si'a": " yeroo "," midhaa": " miidhaa "," dhokaase": " dhukaase "," al": " yeroo ",
                    " ilkee": " ilkaan "," mencaa ": " mancaa "," qaamaa": " qaama "," iubbuu": " lubbuu "
                    " midhamaa": " miidhamaa "," miidhama": " miidhamaa "," miidhaama": " miidhamaa ",
            }

# Regular expression for finding contractions
word_re=re.compile('(%s)' % '|'.join(word_dic.keys()))

# Function for expanding contractions
def normalize(text,word_dic=word_dic):
    def replace(match):
        return word_dic[match.group(0)]
    return word_re.sub(replace, text)

# Expanding Contractions in the title, text
df['case'] = df['case'].apply(lambda text: normalize(text))
```

Figure 5.4: Implementation of word normalization

5.6.3. Implementation of Remove Stop-words

After the implementation of the cleaning dataset and normalization is done, we applied other data preprocessing techniques called remove stop-words. Python programming language can provide an NLTK module that removes stop words from the dataset. Afaan Oromo language stop-words were collected and implemented in this study. Appendix B has shown the list of Afan Oromo stop-words. Figure 5.3 shows the code written to remove the stop words.

5.6.4. Implementation of Tokenization

Implementation of Tokenization is required to get the pattern of the text and obtain the tokens from the dataset. The data collected and prepared is in the form of sentences. Therefore, to make tokenization or convert a text to streams of tokens we have used the NLTK module called *word_tokenize ()* function.

```
#remove stop words
th=df["case"]
sentence=' '.join(th)
stop_words = set(stopwords.words('oromic'))
#word tokenization
word_tokens = word_tokenize(sentence)

filtered_sentence = [w for w in word_tokens if not w.lower() in stop_words]

filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)

print(word_tokens)
print(filtered_sentence)
```

Figure 5.5: Implementation of stop-words and tokenization

5.7. Feature Extraction Implementation

Computer machines only understand or recognize a number. Due to this, we must convert raw text or tokens to vectors or binary numbers through feature extraction techniques. In our study, we implement three features, namely N-gram, TFIDF, and BOW.

5.7.1. Implementation of TFIDF

In this study, TF-IDF features extraction methods were implemented using the sci-kit learns package of *TfidfVectorizer* class. This class converts a case of the dataset to a matrix of TFIDF features vectors. TFIDF contains the word frequency and its importance in the dataset. For feature modeling with TF-IDF, we instantiate the TfidfVectorizer function and pass the case text in the dataset to the *fit_transform ()* method of the function. Various parameters have experimented with TFIDF by changing through the hand, and the best parameter has been selected for the final.

```
#Convert text to vectors using TFIDF
vectorization = TfidfVectorizer(use_idf=True,
                                smooth_idf=False,
                                norm=None, decode_error='replace',
                                max_features=1000,

                               max_df=0.75)
x_FE = vectorization.fit_transform(x)
```

Figure 5.6: Implementation code for TF-IDF

5.7.2. Implementation of BOW

To implement a bag of word feature extraction, the CountVectorize () function of the scikit-learn library has been used. Like TF-IDF feature extraction, a bag of words feature extraction has experimented with different parameters to get the best parameter.

```
#Convert text to vectors using BOW
matrix = CountVectorizer(max_features=1000)
x_bow = matrix.fit_transform(x).toarray()
```

Figure 5.7: Implementation code of BOW feature extraction

5.8. Machine Learning Model Implementation

In this study to implement a machine learning algorithm, we have used the scikit-learn library. After we import the scikit-learn library, we create an object, fitting the model and make a prediction. However, before training or fitting the model we have categorized the vectors dataset into independent and dependent variables. Independent features were merged in one place, then it has been held on the X variable. The y variable also contains the target class or dependent feature. Then the classifiers have been trained the entire data set using X and y. To train a model *fit ()* function was used with appropriate parameters.

```
#splitting dataset to dependent and independent variables
#for binary classification
X=df["case"]
y=df["Gocha"]

#splitting dataset to dependent and independent variables
#for multi class classification
X=df["case"]
y=df["Gulanta"]
```

Figure 5.8: Splitting dataset into dependent and independent for both classifications

***Implementation of Support Vector Machine Model*:** Most of the time support vector machine yields better performance on binary classification. There are different types of SVM classifiers among SVM classifiers, we have implemented an SVC classifier for both classification problems. SVC classifier uses one vs. the rest scheme. That means it has also the ability to handle multi-class using one vs. rest. To use optimal parameters, we have implemented GridSearchCV () with 10 fold stratified cross-Validation, Appendix C1. The following parameters are obtained after the implementation of GridSearchCV ().

```
Parameters that give the best results :

 {'C': 10, 'kernel': 'linear'}


Estimator that was chosen by the search :

 SVC(C=10, kernel='linear')
```

Figure 5.9: Parameters chosen by gridsearchcv for SVM model

***Implementation of Random Forest Model*:** We implemented Random Forest Classifier using *RandomForestClassifier ()* function. This classifier uses a built-in L2 regularization and Decision Tree base estimator. However, to increase the performance of the model, GridSearchCv () was implemented for this classifier. Then, the trained data is saved in the pickle database Appendix C2.

```
{'criterion': 'entropy',
 'max_depth': 8,
 'max_features': 'auto',
 'n_estimators': 500}
```

Figure 5.10: Parameters chosen by gridsearchcv for RF model

***Implementation of Naïve Bayes Model*:** We have been implemented naïve bayes Classifier using *MultinomialNB ()* function. Like the above models, to get the optimal parameters GridSearchCv () hyper-parameters technique has been implemented using 10 stratified cross-validations. The sample code of NB implementation was shown in Appendix C3.

5.9. Implementation of Model Evaluation

We evaluated our models by using 10-fold cross-validation. From the 10 folds, 9 folds are used for training and 1 fold for testing iteratively. In figure below *cross_val_score* () and *cross_val_predict* () methods are used to evaluate our models. *Cross_val_score* is used to calculate the average accuracy of the 10 folds while *cross_val_predict* is used to return the predicted label. In addition to this, we evaluated our model performance using evaluation metrics such as confusion matrix, precision, recall, and F1-score.

CHAPTER SIX

6. EVALUATION, RESULTS, AND DISCUSSIONS

6.1. Introduction

In this chapter, the resulting experiment of the proposed solution for predicting a judicial decision of the Oromia Supreme Court using machine learning is discussed. In addition to these, the results of the dataset, the result of feature extraction, the result of model evaluation, and the effect of removing stop-words and keep stop-words, the result of human evaluation, and the result of judgment (binary classification) and penalty (multi-class classification) model are discussed. Finally, we discussed the significance and the findings of the result obtained by each experiment in this study.

6.2. Dataset Class Distributions Result

The prepared dataset has been divided into judgment and penalty. The judgment dataset has 1638 instances with 6 columns including one target class and the penalty dataset has 868 instances with 8 columns including one target class. For this study, when labeled 1638 total instances of a judgment dataset, 770 and 868 instances are classified as not guilty and guilty respectively. The judgment dataset contains two classes; not guilty and guilty, whereas the penalty dataset contains 38 classes. In the penalty dataset, each class contains 10-25 instances. The details of dataset class distribution for both judgment and penalty are shown in the figure below before apply data imbalance technique.

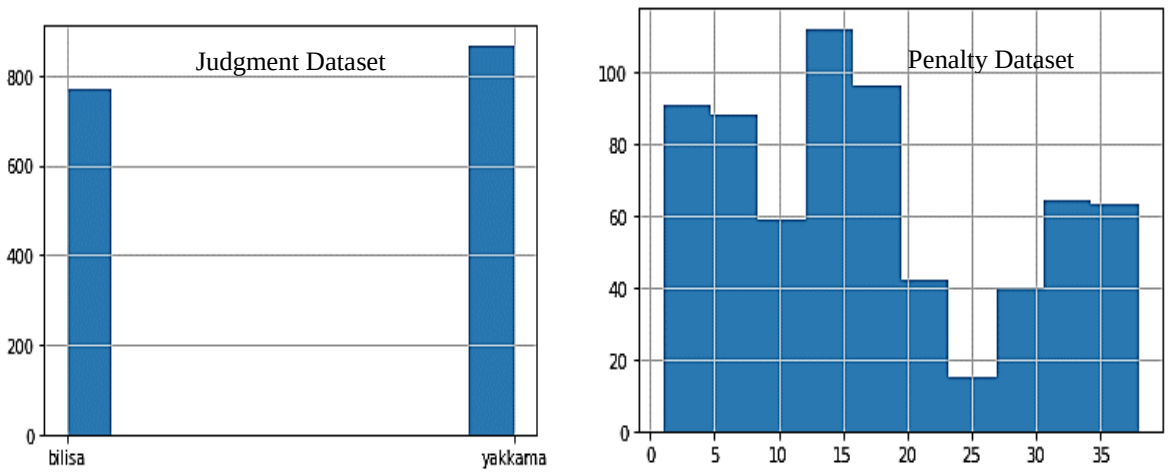

Figure 6.1: Datasets distribution before SMOTE applied

As illustrated in Fig 6.1 shown above, there is a slight class imbalance in the dataset in which instances are not equally distributed among the target classes. When it comes to the machine learning approach, training the model with class imbalance dataset distributions

makes the model to be biased toward the majority class that causes prediction problems toward the minority class, and reduces the overall accuracy of the model. To handle this problem we have been used over-sampling techniques that called SMOTE to balance both judgment and penalty datasets. As described in figure 6.1, the class distribution is imbalanced. SMOTE technique oversampling the minority class and makes the balance or equal with the majority class. It was applied on training dataset. The following figure depicts the balanced data after SMOTE is applied.

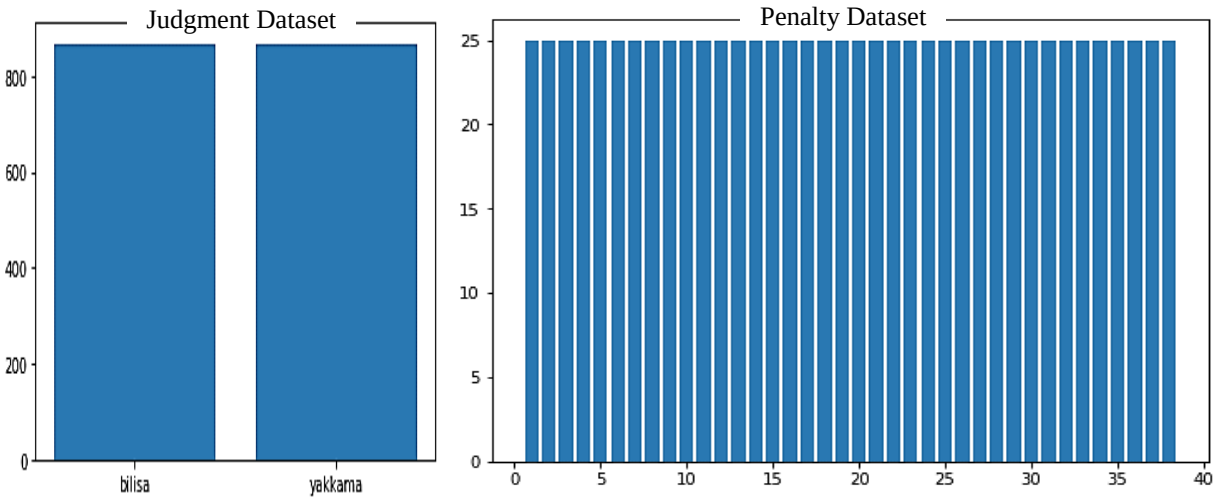

Figure 6.2: Datasets distribution after SMOTE applied

As mentioned in chapter three, the dataset we have been obtained is scanned documents. Therefore, converting a scanned document into textual data is tedious and time-consuming. In general, it took more than four months to prepare, preprocess the dataset and to train the models.

6.3. Model Evaluation Result

As we discussed in the previous chapter, the experimentation of two models: judgment and penalty has been done. In our study, machine learning algorithms such as RF, SVM, and NB with TF-IDF, BOW, and N-gram (unigram, bigram, and trigram) have been implemented for both models separately. In addition to TF-IDF and BOW feature extraction, we have been added the other feature extraction called N-gram and made a comparison with the proposed feature extraction.

Before going to express the result of those models, let us see the result of one experiment that related to NLP. This is about removing stop-words. We tried to identify the importance of stop words by experimenting with those models without removing stop

words and avoiding stop words. Because in judgment and charge writing, every word would be written seriously. However, we have achieved better results by avoiding or removing Afaan Oromo stop words instead of keep stop words from our dataset. The results of feature extractions without remove stop-words are shown in Appendix C4. So, we have been focused on the result of models with removed stop words.

The second one is the result of the cross-validation. To avoid overfitting and see the skill of models on unseen data, we have used cross-validation. In the k-fold, the chances of getting very unbalanced folds are high, which makes model training to be biased. But the stratified CV can overcome these problems and conserves the imbalanced class distribution. For this reason, stratified CV has been used in our study, because we have an imbalanced dataset. But the value of k is not well known; therefore we have trained our model with different k-values (2-10) and select the best among them. Among k values (2-10), 10 fold stratified CV was yield better performance. Due to this, a 10 fold stratified CV has been used in this study. The results of each k-values (2-10) have been shown in Appendix D1 and D2.

The third one is the result of hyper-parameters. To find the optimal parameters to those models we have been used hyper-parameters techniques. As explained in chapter 3 section 3.5.2, there are different types of hyper-parameter algorithms. Of these, grid search and random search results are compared. However, a grid search with stratified 10 fold cross-validations on three models has yielded the best results. The result of randomized search with stratified 10 fold cross-validations of three models is shown in *Appendix C5*. Let us put the aforementioned results for three selected models separately in terms of accuracy, precision, recall, f1-score, and confusion matrix to identify the most performed model.

6.3.1. **Judgment Models Evaluation Results**

The dataset that has a binary class was created to predict the judgment that guilty or not guilty. Using this dataset, we have trained and testing models and evaluated them by using stratified 10 fold CV and other evaluation metrics, like precision, recall, f1-score, and confusion matrix. The result mean or average stratified 10 fold CV accuracy of SVM, NB, and RF models are presented in a table below.

Table 6.1: 10 fold stratified CV of average accuracy for three models

Feature Extraction	*SVM model with 10 Fold Average Accuracy (mean) %*	*RF model with 10 Fold Average Accuracy (mean) %*	*NB model with 10 Fold Average Accuracy (mean) %*
TF-IDF	94.41	**96.02**	93.02
BOW	**93.25**	93.95	**93.78**
Unigram	70.45	68.44	79.36
Bigram	84.26	82.32	81.34
Trigram	91.24	92.54	83.92

The results in Table 6.2 show the Stratified CV average accuracy scores of 10 folds obtained for the SVM, NB, and RF models based on the five feature extractions. TF-IDF feature extraction was yielded 94.41 % and BOW feature extraction yielded 93.25 % of average accuracy or mean accuracy with the SVM model. And also remain feature extraction yielded 70.45 %, 84.26 %, and 91.24 of unigram, bigram, and trigram respectively with the SVM model. To obtain the result of NB shown in the above table, we make many experiments with different types of Naïve Bayes algorithms. However, as depicted in the above, the Naïve Bayes model was yield 93.78 % average accuracy with BOW feature extraction rather than other feature extractions. But it was also yielded 93.02 %, 79.36 %, 81.34 %, and 83.92 % average accuracy with TF-IDF, unigram, bigram, and trigram respectively.

Like the other models, the Random Forest model also yielded better results with the above different feature extractions. RF model obtained a 96.02 % CV average accuracy score with TF-IDF and a 93.95 % CV average accuracy score with BOW feature extraction. RF model also scored 68.44 % with unigram, 82.32 % with Bigram, and 92.54 % with trigram CV average accuracy score.

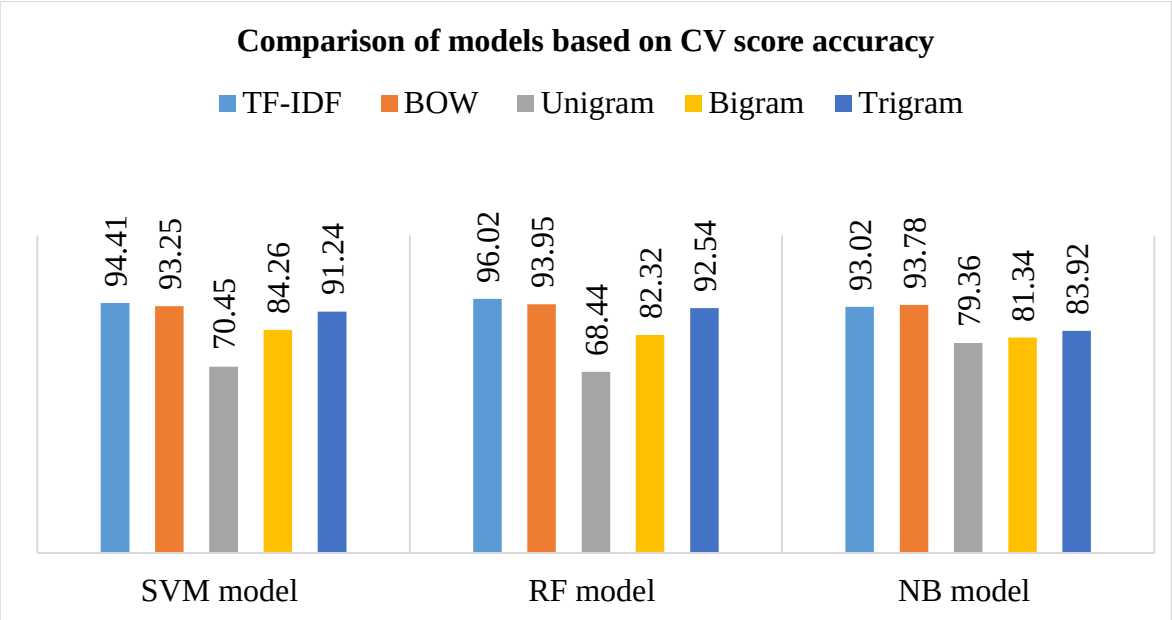

Figure 6.3: A comparison of three models based on the stratified 10 fold CV average accuracy

From the above result that shown in table 6.1 and chart graph 6.3, the feature extractions that have high accuracy were selected among them. TF-IDF and BOW scored better results on all three models than others. The below chart graph depicts the feature extraction that has higher accuracy than the others with three models.

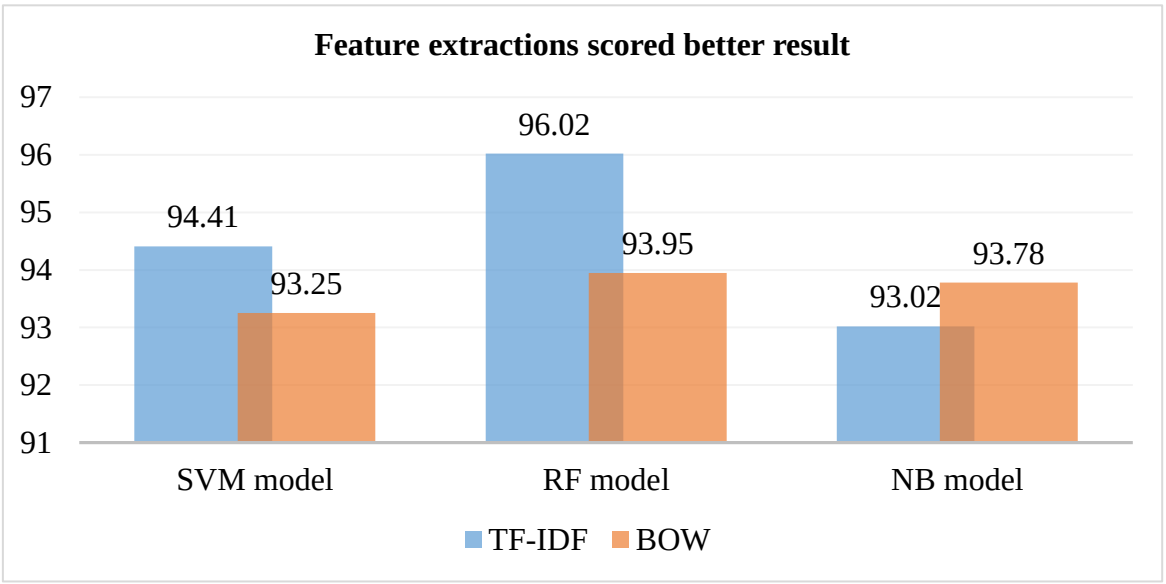

Figure 6.4: Feature extractions scored better result with three models

6.3.1.1. **Hyper-parameter Tuning Results**

The results shown in table 6.1, are the results obtained by default parameters. The following results are the result obtained after the parameter was tuned using grid search.

We have been to make tuning parameters on the three models with selected feature extraction (Figure 6.3) due to their higher performance.

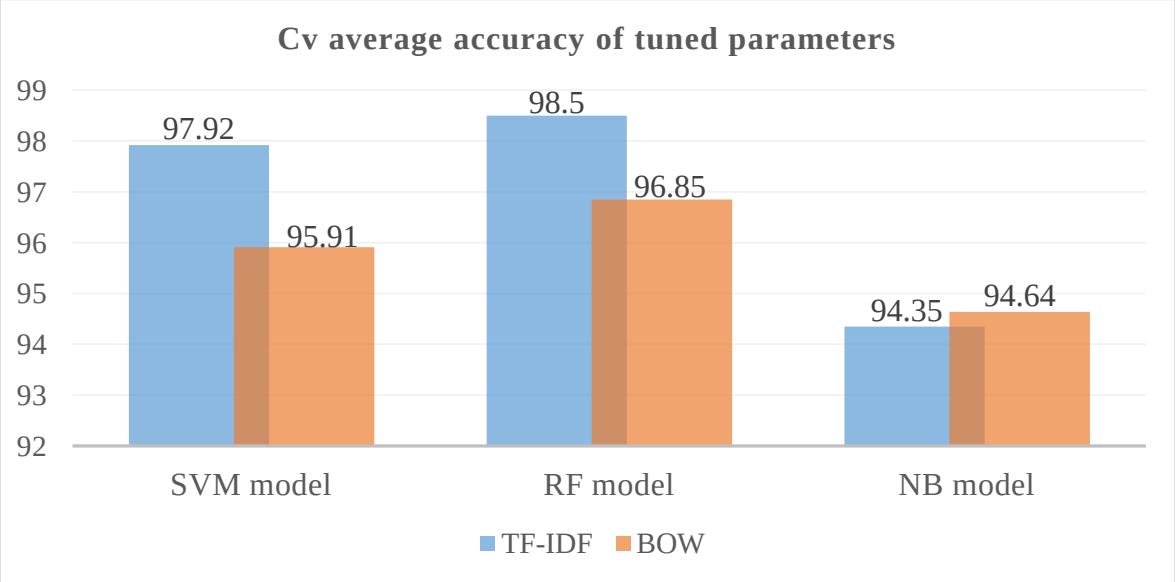

Figure 6.5: Result of hyper-parameter tuning of three models with selected feature extraction

The result of these models is the result obtained after the optimal parameters have been tuned or chosen. After parameter tuned, the SVM, RF, and NB models achieved 97.92 %, 98.5 %, and 94.35 % CV average accuracy, respectively, with the TF-IDF feature extraction rather than the BOW feature extraction. Based on the CV average accuracy obtained, this is a significant result for the judgment model.

6.3.1.2. **Result of Classification Metrics**

In addition to the stratified ten-fold CV score, the study uses the model's performance evaluation metrics as described in Chapter 3 section 3.3. These metrics are Precision (P), Recall (R), and F1-score (F1). The results of these metrics are presented in the following table.

Table 6.2: Result of classification metrics

Feature Extraction	***SVM model in %***			***RF model in %***			***NB Model in %***		
	P	***R***	***F1***	***P***	***R***	***F1***	***P***	***R***	***F1***
TF-IDF	96	98	97	98	98	**98**	94	95	94
BOW	96	97	**96**	96	97	97	95	95	**95**

F1-score is the harmonic mean of precision and recall. Therefore, we make comparative classification metrics based on the F1-score. TF-IDF feature extraction yielded a higher F1-score of 98 % with the RF model, 94 % with the NB model, and 97 % with the SVM model. However, the SVM, NB, and RF models were obtained a 96 %, 95 %, 97 % F1-score with BOW feature extraction respectively.

In precision and recall, if precision is high, the false-positive rate will be low, and also if the recall is high, the false-negative rate will be low. High precision and high recall mean that you have accurate results but if you have low precision and high recall, then it means that most of the predicted values are false.

6.3.1.3. **Result of Confusion Matrix**

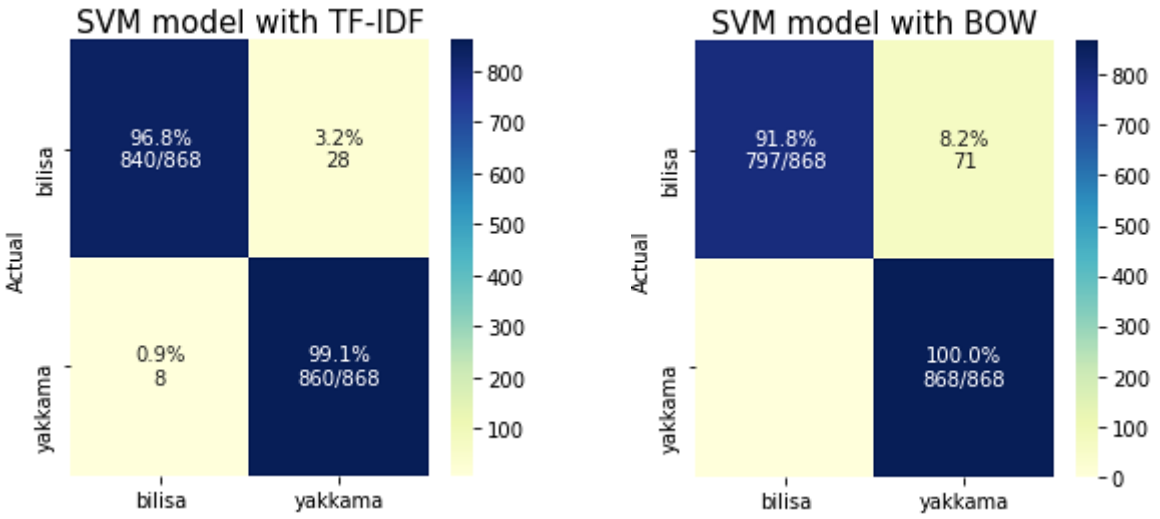

Figure 6.6: Normalized confusion matrix of SVM model with two feature extraction

Figure 6.5 shown the normalized confusion matrix of the SVM model with two feature extraction. SVM classifies 96.8 % of `bilisa` and 99.1 % of `yakkama` correctly and 3.2 % of `bilisa` as `yakkama` and 0.9 % of `yakkama` as `bilisa` are misclassified with TF-IDF feature extraction. However, SVM classifies 91.8 % of `bilisa` and 100 % of `yakkama` correctly and 8.2 % of `bilisa` as `yakkama` are misclassified with BOW feature extraction.

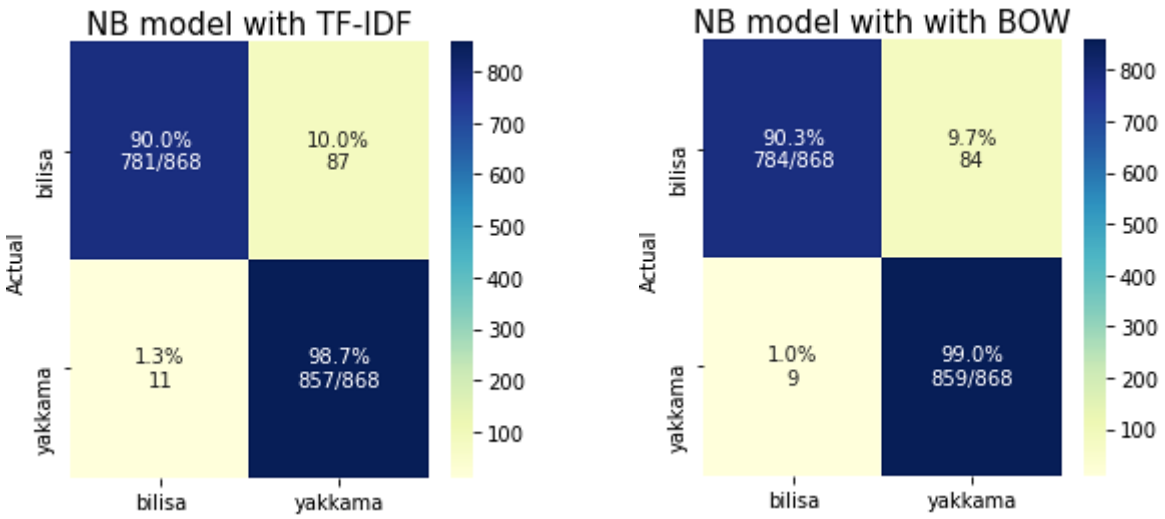

Figure 6.7: Normalized confusion matrix of NB model with two feature extraction

As shown in Figure 6.6 NB classifies 90.0 % of `bilisa` and 98.7 % of `yakkama` correctly. NB misclassify 10.0 % of `bilisa` as `yakkama` and 1.3 % of `yakkama` as `bilisa` with TF-IDF feature extraction. However, it misclassifies 9.7 % of `bilisa` as `yakkama` and 1.0 % of `yakkama` as `bilisa`, as well as it also classifies 90.3 % of `bilisa` and 99.0 % of `yakkama` correctly with BOW feature extraction.

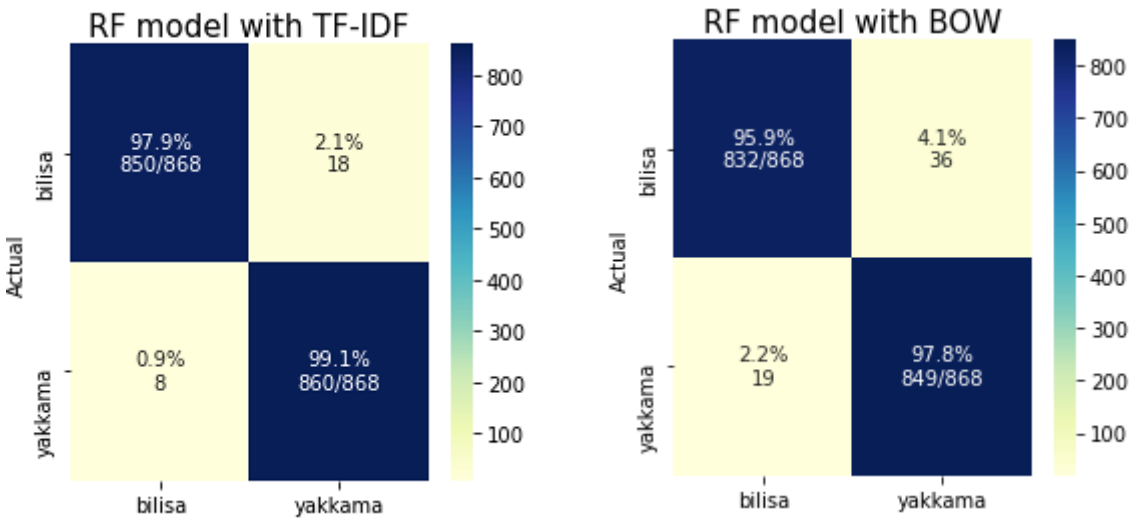

Figure 6.8: Normalized confusion matrix of RF model with two feature extraction

As depicted in Figure 6.7, the RF model classifies 97.9 % of `bilisa` and 99.1 % of `yakkama` correctly and 2.1 % of `bilisa` as `yakkama` and 0.9 % of `yakkama` as `bilisa` are misclassified with TF-IDF feature extraction. However, it classifies 95.9 % of `bilisa` and 97.8 % of `yakkama` correctly and 4.1 % of `bilisa` as `yakkama` and 2.2 % of `yakkama` as `bilisa` are misclassified with BOW feature extraction.

RF model classified the class correctly with TF-IDF rather than SVM and NB models for judgment dataset. Finally, the result of judgment models after chosen optimal parameters demonstrates that the RF model shows better accuracy and f1_score than the SVM and NB models with TF-IDF feature extraction.

6.3.2. **Penalty Models Evaluation Results**

As we discussed in chapters three and four, we have set up a multi-class dataset to predict the punishment of criminals. Using this dataset, we have trained and testing models and evaluated them by using a 10 fold stratified CV. In addition to this, we also used other evaluation metrics, like precision, recall, f1-score, and confusion matrix. The result mean or average accuracy of each test for SVM, NB, and RF models are presented in table 6.4, based on the obtained vectors.

Table 6.3: 10 fold stratified CV of average accuracy for three models

Feature Extraction	***SVM model with 10 Fold Average Accuracy (mean) %***	***RF model with 10 Fold Average Accuracy (mean) %***	***NB model with 10 Fold Average Accuracy (mean) %***
TF-IDF	**77.98**	**74.28**	61.89
BOW	72.66	73.95	70.42
Unigram	59.89	60.48	56.81
Bigram	61.67	64.72	58.29
Trigram	71.92	72.33	59.97

The result shown in Table 6.3 is the result obtained by default parameters. Through default parameters, the SVM model with TF-IDF, and RF model with TF-IDF yielded higher results, that 77.98 % and 74.28 % CV average accuracy, respectively. As well as the NB model obtained a 70.42 % higher score with BOW. All three models scored lower with unigram, Bigram, and Trigram feature extraction. But, they also recorded higher scores with TF-IDF and BOW feature extraction. In other words, TF-IDF and BOW feature extraction yielded higher CV average accuracy on all models.

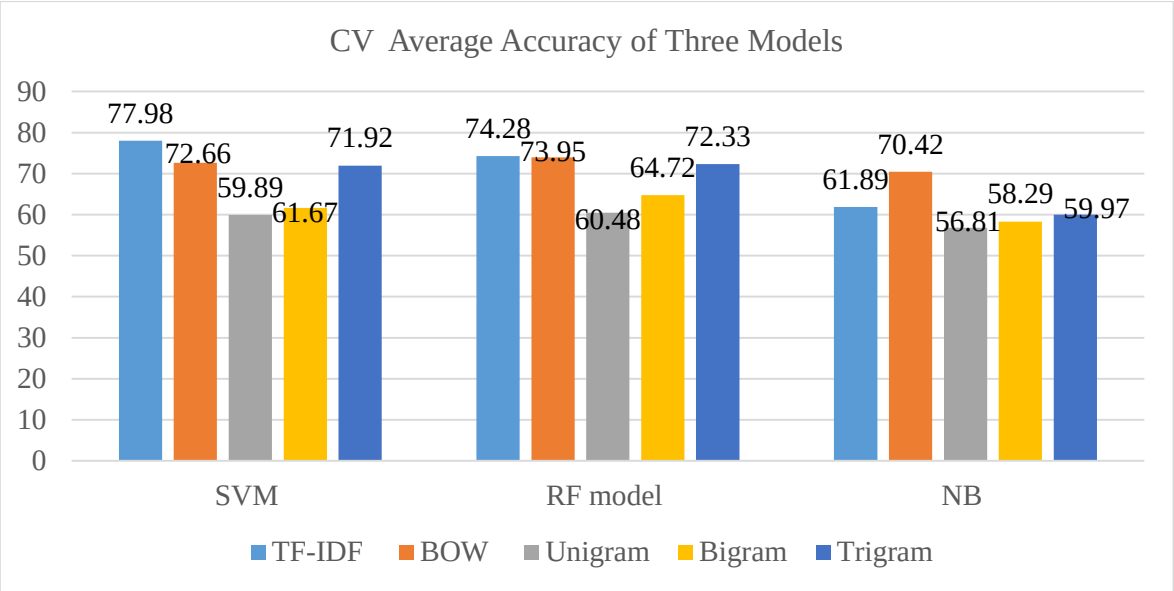

Figure 6.9: A comparison of three models based on the stratified 10 fold CV average accuracy of penalty models

6.3.2.1. **Hyper-parameter Tuning Results**

The results shown in Table 6.3, are the results obtained by default parameters. We used hyper-parameter tuning algorithms to find the best parameters. The following results are the result obtained after the hyper-parameter was tuned using grid search. To tune the parameters of our models, we have selected only two feature extraction (TF-IDF and BOW) that has higher accuracy on the default parameters.

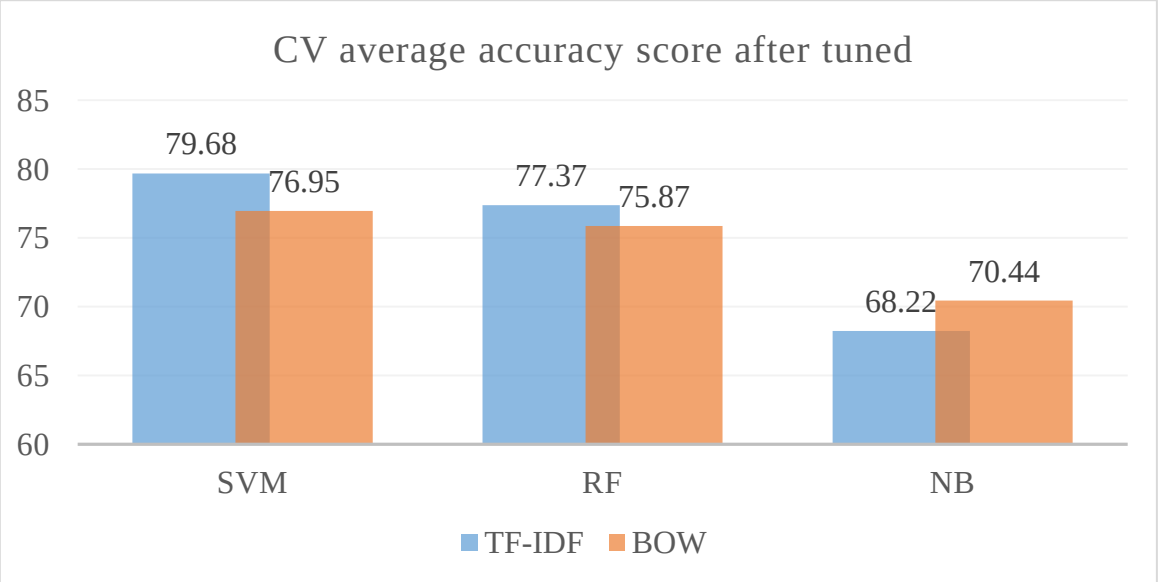

Figure 6.10: Result of hyper-parameters with TF-IDF and BOW

As shown in figure 6.10, the SVM model obtained 79.68 % with TF-IDF and 76.95 % with BOW CV average accuracy after optimal parameters were selected. RF model yielded

77.37 % with TF-IDF, and 75.87 % with BOW after parameters were tuned. NB model also obtained 68.22 % and 70.44 % with TF-IDF and BOW after tuned parameters.

6.3.2.2. **Result of Classification Metrics**

As mentioned in the previous chapters, other classification metrics were also used to evaluate the proposed model. These metrics are Precision (P), Recall (R), and F1-score (F1). The result of those metrics was depicted in the following table.

Table 6.4: Result of classification metrics

Feature Extraction	***SVM model in %***			***RF model in %***			***NB Model in %***		
	P	***R***	***F1***	***P***	***R***	***F1***	***P***	***R***	***F1***
TF-IDF	78	80	**79**	79	76	**77**	64	69	66
BOW	75	77	76	74	76	75	66	70	**68**

Based on the F1-score, the SVM model obtained 79 % with TF-IDF and 76 % BOW feature extraction. If a high recall and low precision, then it means that most of the predicted values are false. But if they are close, the predicted value will be good.

RF model obtained 77 % of F1-score with TF-IDF and 75 % of F1-score with BOW. And also, the NB model yielded 65 % of the F1-score with BOW and 68 % of the F1-score with TF-IDF feature extraction. Generally, the SVM model yielded a higher result than other models.

6.4. Result of Human Evaluation

As we discussed in chapter three, the model with the best results for deploying the prototype was selected. We have integrated the model into the web using a flask server to evaluate the model through human evaluation. Firstly we designed a user interface using HTML, CSS, and JavaScript (Appendix E0). This model was evaluated by six law experts. One judge and law officers from OSC, and also one judge and law officers from the high court, as well as one judge and law officers from first instance court. The experts then reviewed the model using the evaluation sheets provided in *Appendix E1*. Those experts focused primarily on model performance evaluation. This model performance evaluation is based on the number of cases predicted correctly by the proposed model. We used the following formula to calculate the correctness of the proposed model.

$$Correctness = \frac{total\ number\ of\ cases \text{ accurately predicted}}{total\ number\ of\ cases \text{ provided for the model}} * 100 \quad (6.1)$$

Table 6.5*:* The human evaluation results on model performance

Number of persons	***Total number of entered cases***	***Total number of cases predicted correctly***	***The total number of cases predicted not correctly***	***Correctness in %***
OSC law experts (2)	20	15	5	75%
High court law experts (2)	14	12	2	85.71%
First court law experts (2)	6	4	2	66.6%
Total	40	31	9	**77.5%**

As we have seen in the above table, the performance of the model is evaluated by the different law experts based on the number of cases that experts are provided for the model and the total number of cases that the model accurately predicted. According to the law expert's evaluation, our proposed model is correctly predicted at 77.5%, and only 22.5% is not correctly predicted.

6.5. Discussion

To our knowledge, this study is the first to predict the judicial decision of the Oromia Supreme Court using a machine learning technique based on Ethiopian law and the case prepared by the Oromo language. We have conducted several experiments on the newly collected dataset and have found promising results. In this study, we examined different machine learning with different feature extraction. And also the importance of removing stop-words and keep stop-words in our dataset was examined. As summarized in Figures 6.5, 6.10 and table 6.2 and 6.4, the RF and SVM model performed better in terms of f1-score, accuracy, and precision to predict judgment and penalty, respectively. Generally, our results confirm obtain good results that separating datasets into judgment and penalty and training the RF and SVM model with tuned parameters by removing stop-words and by extracting features through TF-IDF feature extraction.

In addition, in this section, we discuss how this study answers the research questions raised in chapter one Section 1.4. We have been used a variety of techniques to answer these

questions. Let's look at the techniques used to answer research questions and how the proposed model answered those questions one by one.

RQ1: How precisely can the judgments made by the Oromia Supreme Courts be forecasted with punishment using machine learning? To answer this question, different techniques and methods were employed. The first technique is reviewed and examined how the actual judicial decision would be made by judges. A judge gives a verdict or judicial decision based on the Ethiopia Criminal Code Procedure (ECCP). ECCP, identify which factors are very important in judicial decisions to give a judgment. Therefore, to train the machine learning model we have been prepared the dataset according to the ECCP in order to give an appropriate prediction. Then, we have been separated the dataset into judgment and penalty. The second technique is cleaning datasets to make them more suitable for ML models by using data preprocessing techniques.

RQ2: Which machine learning models and feature extraction are most suitable for judicial decision prediction? To answer this question, firstly we set out criteria to selected machine learning model and feature extraction. After we selected model and feature extraction based on these criteria, we have been trained and test models using the prepared dataset. To identify best-performed models, model performance evaluation (stratified 10 fold CV), classification metrics (accuracy, precision, recall, and f1-score), and human evaluation would be used. Our empirical analysis indicates that the RF model with TF-IDF for judgment and SVM models with TF-IDF for penalty prediction is best.

RQ3: How can we improve the quality of judicial decision predictions? In addition to preparing the dataset and select models, finding hyper-parameters techniques that were used to optimize our models is very important. In this, grid search and randomize hyper-parameters techniques were compared to find optimal parameters. To identify this, first, we have examined the model with default parameters. However, the results obtaining after tuning using grid search were better than default parameters.

As we discussed in chapter two section 2.3, several related studies have been conducted using machine learning and other NLP techniques to predict judicial decisions. We have selected three studies that are closely related to our study for comparison.

In this study (Aletras et al., 2016), the researcher was predicted only judgment (violation and non-violation) using a dataset with 584 instances obtained from an online repository.

The author first separates this dataset based on their type of article (article 3, 6, 8) and assigns a violation and non-violation dataset for those articles. N-gram and topic model was used to train SVM classifiers. The researcher was obtained 79 % of accuracy.

The second study is the study done by (Medvedeva et al., 2020). The researcher's work is very similar to Nikolas Alteras, el., The study focused on the improvement of Nikolas Alteras, el, work, but the researcher has increased the number of data sets to 3132 and articles to 9. SVM classifiers were used to classify the data to violation and non-violation with 75 % of accuracy.

The third study is the study done by (Shaikh et al., 2020). The researcher tried to predicting a judgment (acquittal or conviction). For this study, the researcher has used the dataset have 86 instances that related to murder case. The researcher was compared different machine learning algorithms (LR, KNN, CART, NB, SVM, Bagging, RF, and Boosting) based on their accuracy and F1-score. But the researcher did not explicitly put which types of feature extraction were used. However, obtained 91.86 % accuracy with CART.

In our proposed solution, we implement different techniques to predict judicial decisions. When we are saying judicial decision, it contains two main things: judgment and penalty or punishment. Judgment is the first part of the trial, which determines whether the suspect is guilty or not. Punishment is a penalty imposed upon an accused who is guilty. Before we can punish, we must determine whether the defendant is guilty. In this study, we are tried to seem the work way of our proposed model with a real-world judicial decision procedure. We are prepared or arranged our dataset based on the court procedures. Court procedure (Imperial Ethiopian Government, 1969) tells which factor or term plays a great role in judicial decisions as discussed in chapter two section 2.4.1. After the dataset is prepared, we implemented different NLP techniques. To identify the importance of stop-words in our dataset, we have been trained our model with removing and without removing Afaan Oromo stop-words from our dataset.

In this study, different models like RF, SVM, and NB are trained with N-gram, TF-IDF, and BOW feature extraction individually. In order to find optimal parameter grid search was used. To evaluate their performance stratified 10 fold cross-validation was used in addition to classification metrics like accuracy, precision, recall, f1-score, and confusion matrix. In addition to those performance evaluation techniques, we also used human evaluation methods.

Table 6.6: Comparison of our model with previous work

Author	*Methodology*		*# of instance in Dataset*	*Target*		*A model with the best accuracy % result*
	Model	*Feature Extraction*		*Predict Judgment*	*Predict Penalty*	
Nikolas Alteras el (2016)	Only SVM	N-gram	584	Yes	No	SVM with 79 % of accuracy
Medvedeva (2018)	Only SVM	TF-IDF	3132	Yes	No	SVM with 75 % of accuracy
Rafe Athar Shaikha (2019)	CART, KNN, LR, RF, and Bagging	Not clearly put	86	Yes	No	CART with 91.86 % of accuracy
Proposed model	SVM, RF, and NB	N-gram, TF_IDF, BOW	1638	Yes	Yes	RF 98.5 % of accuracy for judgment and SVM 79.68 % of CV accuracy for Penalty with TF-IDF

6.6. CONCLUSION, CONTRIBUTION, AND FUTURE WORK

6.6.1. **Conclusion**

Predicting legal decisions is used to support judges, lawyers, and non-professional of law to know the outcome of cases. In order to help or support them, we have been proposed a judicial decision prediction model. The general objective of this study is to predict the judicial decision of OSC using machine learning. We predict the judgment results from two aspects: the judgment (accusation) and the penalty. To do this we have been used a different technique.

The first technique is building a dataset. Since there was no previous resource (dataset) available about the area we wanted to explore, we used a new dataset. To build a new dataset, we collected criminal case documents from OSC that related to murder and injury bodies. OCR, (MS OneNote) software was used to convert scanned documents into text data because the collected document is a scanned file. Then after, we implemented different data preprocessing techniques to make more suitable dataset for the ML algorithm.

The computer cannot understand the raw text. Once the dataset is ready, we have been used the feature extractions to convert the raw text into a computer-readable form (number). We make the comparison between feature extractions, such as N-gram (unigram, bigram, and trigram), BOW, and TF-IDF in order to select the best one. To train models, the researcher applied three different classification techniques, such as SVM, RF, and NB models. Firstly, those models are selected based on some criteria as discussed in section 3.6. Grid search hyper-parameter tuning technique was implemented on all three models to find the optimal parameter. The result of those models was evaluated by 10 fold stratified cross-validation.

Our results confirm obtain good results by predicting a judgment using the RF model and predict a penalty using the SVM model, on tuned parameters with TF-IDF feature extraction. After the best-performed model selected, we have been integrated the model to the web by using the Flask framework for two purposes. The first purpose is should make it accessible to users. The second purpose is to evaluate the models by law experts in addition to evaluating models by classification metrics. This human evaluation was done by six law experts. The law experts evaluated the model simply by inserting new cases. They evaluated the model with 40 new cases, and the model correctly predicted 31 cases

out of 40 cases. That means the proposed model obtained 77.5 % of accuracy through human evaluation.

6.6.2. **Recommendation**

The prevalence of machine learning in the legal or law domain is low, especially in Ethiopian law. In addition to these, many problems are facing the Ethiopian courts. So, we recommend if other researchers have done their study on Ethiopian law using different algorithms that to solve the problem that faced the Ethiopian courts and as well as to modernize the courts. As we have seen in the results of the model evaluation, this study has received good human evaluation results from law experts. So, if we turn this study into a project, it will reduce the burden on judges and create more efficient time. We recommend that the Oromia Supreme Court see this study and turn it into a project.

6.6.3. **Contribution**

We tried different techniques and approaches to address the problem raised in chapter one in section 1.3 and to answer the research questions. In this study, to obtain this result experiments were done. In this study, we contributed new or additional things and also improved the quality of prediction. The main contribution is:

- We have created a new collection of judgment datasets: in machine learning, a dataset is very important. Previously, there is no judgment dataset prepared by the Ethiopian language. So, to create a new dataset we have been used different techniques. Firstly, we converted the scanned case documents to text data using MS OneNote. The created dataset contains judgment and punishment information. This dataset was created based on the Ethiopian Criminal Court procedure. We have been tried to mimic the way work of our proposed model with the judge's work.
- Predicting the punishment of the guilty: if a defendant is convicted, the judge imposes the punishment. Impose a punishment for a judge is very simple. But, it’s very difficult for a computer machine. However, in our study, we have been finding the way how the developed models are predicting the punishment of the guilty.
- Improving the quality of prediction in the judicial decision: to improve the quality of prediction we have been used two techniques. The first technique is to identify which factors are very important in judicial decisions based on the Ethiopian Court procedure. The second technique is to use a hyper-parameter tuning technique.

6.6.4. **Future Work**

In this study, to build the prediction judicial decision model we used a supervised learning algorithm with a feature extraction method. In addition to this, we have implemented a variety of data preprocessing techniques. There are a variety of techniques that we have not implemented, such as lemmatization and stemming data preprocessing techniques and ML algorithms due to time constraints. It is best to look at the differences in performance results using deep learning or unsupervised algorithms models and feature engineering techniques. In addition to this, there are different types of law in the Ethiopian legal system called civil law, criminal law, tax law, sale law, and torture law. In this study, we have included only criminal law. Because the collected document is a scanned document, it needs OCR to convert the scanned document to text data. This was tedious and time-consuming. Therefore, future work focuses on incorporating the rest of the law.

* * *

Special Acknowledgement: This research was funded by Adama Science and Technology University under the grant number **ASTU/SM-R/273/12**.

Adama, Ethiopia.

7. REFERENCES

Abate, T. (2014). *Introduction to Law and the Ethiopian Legal System*. *272*, 640.

Abdi, G. (2016). Oromia Criminal Adjudication by State Courts under the FDRE Constitution: The Quest for Compartmentalization of Jurisdiction. *Joornaalii Seeraa Oromiyaa*, 1–35.

Abinash Tripathy and Ankit Agrawal and Santanu Kumar Rath. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Syst. Appl.*, *57*, 117–126.

Adrian Erasmus. (2010). *Introduction to Support Vector Machines, http://www.svms.org/introduction.html*. *2011*(January 1st), 1. http://www.svms.org/introduction.html

Agrawal, T. (2021). Hyperparameter Optimization in Machine Learning. In *Hyperparameter Optimization in Machine Learning* (1st ed.). Apress. https://doi.org/10.1007/978-1-4842-6579-6

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*, *2016*(10), 1–19. https://doi.org/10.7717/peerj-cs.93

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). *A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques*. 1–13. http://arxiv.org/abs/1707.02919

Allibhai, E. (2018). *Hold-out vs. Cross-validation in Machine Learning*. https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f

Asafa, J. (2010). The present and future of the Oromoo people. *Journal of the Oromoo Literature, a Scholarly Publishing Initiatives*, 24.

Ashley, K. D. (2017). Artificial intelligence and legal analytics: New tools for law practice in the digital age. In *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press. https://doi.org/10.1017/9781316761380

Berrar, D. (2018). Bayes' theorem and naive bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, *1–3*(January 2018), 403–412. https://doi.org/10.1016/B978-0-12-809633-8.20473-1

Brownlee, J. (2018). *k-fold Cross-Validation*. https://machinelearningmastery.com/k-fold-cross-validation/

Brownlee, J. (2020). *Types of Classification Tasks in Machine Learning*. https://machinelearningmastery.com/types-of-classification-in-machine-learning/

Cavnar, W. B., & Trenkle, J. M. (2001). N-Gram-Based Text Categorization N-Gram-Based Text Categorization. *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, *May*, 1–14.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*(Sept. 28), 321–357. https://arxiv.org/pdf/1106.1813.pdf%0Ahttp://www.snopes.com/horrors/insects/telamonia.asp

Duncan, M. (2021). *history of the judiciary*. Online. https://www.judiciary.uk/about-the-judiciary/history-of-the-judiciary/

Duresa, D. (2016). *Large Vocabulary Continuous Speech Recognition System for Afaan Oromo using Hidden Markov Model (HMM) Large Vocabulary Continuous Speech Recognition System for Afaan Oromo using Hidden Markov Model (HMM)*. Unpublished" Master's Thesis. Thesis Submitted to ASTU, 107 pp.

Dwivedi, R. (2020). *How Does SVM Algorithm Works*. https://www.analyticssteps.com/blogs/how-does-support-vector-machine-algorithm-works-machine-learning

Eklund, M. (2018). Comparing Feature Extraction Methods and Effects of Pre-Processing Methods for Multi-Label Classification of Textual Data. *Degree Project Computer Science and Engineering*, 11.

Eskinder Mesfin. (2009). *Application of Multilayer Feed Forward Artificial Neural Network Perceptron in Prediction of Court Case's Time Span: The Case of Federal Supreme Courts* (Vol. 2009, Issue 75). Addis Abeba University: Unpublished Master's Thesis.

F.Y, O., J.E.T, A., O, A., J. O, H., O, O., & J, A. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, *48*(3), 128–138. https://doi.org/10.14445/22312803/ijctt-v48p126

Greenleaf, G. (1989). Legal expert systems – robot lawyers? *Computers and Law Newsletter*, *2*, 21–24.

Hussain Mujtaba. (2020). *Types of Cross Validation*. https://www.mygreatlearning.com/blog/cross-validation/

Imperial Ethiopian Government. (1969). *Criminal Procedure Code Of Ethiopia PROCLAMATION No.185 OF 1961* (2nd ed.). authority of the Ministry of Pen.

J, S. Mk. Ac. Ah. (2020). *Data Preparation for Machine Learning - Data Cleaning, Feature Selection, and Data* (1st ed.).

Johnson, D. (2020). *OneNote*. https://africa.businessinsider.com/tech-insider/what-is-onenote-how-microsofts-note-taking-app-can-help-you-organize-your-work/2yy7hgz

Kamiri, J., & Mariga, G. (2021). Research Methods in Machine Learning: A Content Analysis. *International Journal of Computer and Information Technology(2279-0764)*, *10*(2), 78–91. https://doi.org/10.24203/ijcit.v10i2.79

Karniol-tambour, O. (2013). *Learning Multi-Label Topic Classification of News Articles*. 1–6. http://cs229.stanford.edu/proj2013/ChaseGenainKarniolTambour-LearningMulti-LabelTopicClassificationofNewsArticles.pdf

Kedia, A., & Rasu, M. (2020). *Hands-On - Python Natural Language Processing* (1st ed.). Packt Publishing Ltd.

Lage-Freitas, A., Allende-Cid, H., Santana, O., & de Oliveira-Lage, L. (2019). *Predicting Brazilian court decisions*. 1–4. http://arxiv.org/abs/1905.10348

Lawlor, R. C. (1963). What Computers Can Do: Analysis and Prediction of Judicial Decisions. *American Bar Association Journal*, *49*(4), 337–344.

Li, J., Zhang, G., Yan, H., Yu, L., & Meng, T. (2018). A Markov logic networks based method to predict judicial decisions of divorce cases. *Proceedings - 3rd IEEE International Conference on Smart Cloud, SmartCloud 2018*, *1*, 129–132. https://doi.org/10.1109/SmartCloud.2018.00029

Liashchynskyi, P., & Liashchynskyi, P. (2019). *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*. *2017*, 1–11. http://arxiv.org/abs/1912.06059

Liu, Q., & Wu, Y. (2012). Encyclopedia of the Sciences of Learning. *Encyclopedia of the Sciences of Learning*, *April*. https://doi.org/10.1007/978-1-4419-1428-6

Loevinger, L. (1963). Jurimetrics: The Methodology of Legal Inquiry. *Law and Contemporary Problems*, *28*(1), 5. https://doi.org/10.2307/1190721

M, H., & M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201

Marr, B. (2018). *How AI And Machine Learning Are Transforming Law Firms And The Legal Sector*. Forbes. https://www.forbes.com/sites/bernardmarr/2018/05/23/how-ai-and-machine-learning-are-transforming-law-firms-and-the-legal-sector/?sh=5ba1091832c3

Medvedeva, M., Vols, M., & Wieling, M. (2020). Using machine learning to predict decisions of the European Court of Human Rights. *Artificial Intelligence and Law*, *28*(2), 237–266. https://doi.org/10.1007/s10506-019-09255-y

Microsoft. (2021). *OCR in OneNote*. https://support.microsoft.com/en-us/office/copy-text-from-pictures-and-file-printouts-using-ocr-in-onenote-93a70a2f-ebcd-42dc-9f0b-19b09fd775b4

Murphy, E. F., & Plucknett, T. F. T. (1957). A Concise History of the Common Law. *The American Journal of Legal History*, *1*(3), 259. https://doi.org/10.2307/844567

Muskan Kothari. (2020). *Feature Extraction Techniques – NLP*. https://www.geeksforgeeks.org/feature-extraction-techniques-nlp/

Negesse, F. (2015). Classification of Oromo Dialects : A Computational Approach. *Journalism and Communication*, *7*, 1–10.

Oromia Supreme Court. (2019). *Oromia Courts Mission, Objective and Values*. https://oromiacourt.org/en/oromia-courts-mission-objective-and-values.

Parliamentary Counsel. (2013). *Constitution of Queensland*. *August*, 1–69. https://www.legislation.qld.gov.au/view/pdf/inforce/current/act-2001-080

Partner, M., Law, K. I. T., Wales, T. L. S. of E. and, Society, T. L., States, M., Pratt, G. A., Bates, B. L., Rights, H., The, I., Arab, T., To, S., HM Treasury, Financial Conduct Authroity, B. of E., Fisch, C., Meoli, M., Vismara, S., & Kemp, R. (2018). HORIZON SCANNING forward thinking Artificial Intelligence and the Legal Profession. *Economics of Innovation and New Technology*, *29*(November), 51–60. http://www.kempitlaw.com/wp-content/uploads/2018/06/Legal_Aspects_of_AI_Kemp_IT_Law_v2.0_Sept_-2018.pdf%0Ahttps://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/752070/cryptoassets_taskforce_final_report_final_web.

Pound, R. (1923). The Theory of Judicial Decision . III . A Theory of Judicial Decision for Today Author (s): Roscoe Pound Source : Harvard Law Review , Vol . 36 , No . 8 (Jun ., 1923), pp . 940-959 Published by : The Harvard Law Review Association Stable URL : http://. *Havard Law Review*, *36*(8), 940–959.

Purva Huilgol. (2020). *Precision vs. Recall*. https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/

Randall Lesaffer, J. A. (2009). *European legal history : a cultural and political perspective*. Cambridge University Press.

Raschka, S. (2018). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. 49. http://arxiv.org/abs/1811.12808

Rong, X. (2014). *word2vec Parameter Learning Explained*. 1–21. http://arxiv.org/abs/1411.2738

Shaikh, R. A., Sahu, T. P., & Anand, V. (2020). Predicting Outcomes of Legal Cases based on Legal Factors using Classifiers. *Procedia Computer Science*, *167*(2019), 2393–2402. https://doi.org/10.1016/j.procs.2020.03.292

Singh, D. (2019). *Predictive Model Performance Evaluation*. https://medium.com/@divyacyclitics15/what-is-predictive-model-performance-evaluation-8ef117ae0e40

Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, *36*(2), 111–133. https://doi.org/10.1111/j.2517-6161.1974.tb00994.x

Strickson, B., & De La Iglesia, B. (2020). Legal Judgement Prediction for UK Courts. *ACM International Conference Proceeding Series*, 204–209. https://doi.org/10.1145/3388176.3388183

Suddarth & Koor. (2018). *What Are The Types of Criminal Law?* https://suddarthandkoor.com/types-of-criminal-law/

Techopedia. (2020). *Microsoft Excel*. https://www.techopedia.com/definition/5430/microsoft-excel?

Tesfaye, D. (2010). *Designing a Stemmer for Afaan Oromo Text : A Hybrid Approach*. Unpublished" Master's Thesis. Thesis submitted to AAU, 127 pp.

Thompson, I. (2021). *About World Languages*. http://aboutworldlanguages.com/oromo

Tutorial Points. (2017). Python Programming Language Tutorial. *Organizational Behavior*, 1–305.

Tutorial Points. (2020a). *Naïve Bayes Classifier Algorithm*. https://www.javatpoint.com/machine-learning-naive-bayes-classifier

Tutorial Points. (2020b). *Random Forest Algorithm*. https://www.javatpoint.com/machine-learning-random-forest-algorithm

Vidhya, A. (2015). *Simple Guide to Logistic Regression in R and Python*. https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression

Vilchyk, T. (2018). Duties of a lawyer to a court and to a client. *Russian Law Journal*, *6*(4), 62–99. https://doi.org/10.17589/2309-8678-2018-6-4-62-99

Virtucio, M. B. L., Aborot, J. A., Abonita, J. K. C., Avinante, R. S., Copino, R. J. B., Neverida, M. P., Osiana, V. O., Peramo, E. C., Syjuco, J. G., & Tan, G. B. A. (2018). Predicting Decisions of the Philippine Supreme Court Using Natural Language Processing and Machine Learning. *Proceedings - International Computer Software and Applications Conference*, *2*(July 2018), 130–135. https://doi.org/10.1109/COMPSAC.2018.10348

Visentin, A., Nardotto, A., & Osullivan, B. (2019). Predicting judicial decisions: A statistically rigorous approach and a new ensemble classifier. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *2019-Novem*, 1820–1824. https://doi.org/10.1109/ICTAI.2019.00275

Walton, D. (2005). *Argumentation Methods for Artificial Intelligence and Law* (1st ed.). Springer Netherlands.

Waykole, R. N., & Thakare, A. D. (2018). *International Journal of Advance Engineering and Research A REVIEW OF FEATURE EXTRACTION METHODS FOR TEXT*. *Vdv*, 351–354.

wikibooks. (2021). *Alphabet and Pronunciation*. https://en.wikibooks.org/wiki/Afaan_Oromo/Alphabet

Wikipedia. (2021). *Oromo_language*. https://en.wikipedia.org/wiki/Oromo_language

8. APPENDICES

Appendix A: Normalization Words

Table A.1: List of normalized words

Otuu=osoo	"kessaa ": "keessaa"
Otoo=osoo	"marsaa ": "yeroo"
Fundura=fulduraa	"dhaqabsiiseef ": "geessiseen"
Fuula=fula	"himatamtertii ": "himatamtee jirti"
Ishii=ishee	"himatamteertii ": "himatamtee jirti"
Of-eeggannoo =of eeggannoo	"himatamera": "himatamee jira"
Dhakaa=dhagaa	"himatameera": "himatamee jira"
Dhaka=dhagaa	"iddoo": "bakka"
Diree=waraane	"irraa": "gubbaa"
"gayee ": "gahee"	"irra": "gubbaa"
"hiija": "haalo"	"tan": "kan"
"hiijaa ": "haalo"	"si'a": "yeroo"
"yenna ": "yeroo"	"irraa": "gubbaa"
"tara ": "yeroo",	"dhokaase": "dhukaase"
"mencaa": "mancaa"	"al": "yeroo"
"ilkee": "ilkaan"	

Appendix B: Afaan Oromo Stop-words

Table B.1: List of stop-words

isin	eega	yeroo	koo
illee	kun	teenya	tanaafuu
itti	kiyya	gama	isaaf
narraa	sun	akkasumas	kanaafi
keessatti	natti	kanaafuu	aanee
yookaan	keenya	kee	ittuu
ammo	tanaaf	isatti	sitti
dura	booddee	iseen	nu
eegasii	alatti	ani	utuu
ishii	henna	amma	ala
ishiirraa	kan	duuba	yoo
booda	immoo	silaa	isii
nuti	keessa	ati	an
ta'ullee	gidduu	yommuu	tun
bira	yoom	isiin	sana
keessan	jara	akka	irra
na	nurraa	inni	ishiif
kana	siin	isaa	malee
hanga	fi	isaanirraa	
akkuma	kanaaf	hogguu	
naaf	keenna	waan	
gubbaa	jala	warra	

Appendix C1: Implementation of Gridsearch on SVM Model with 10 Fold Stratified CV

```
# instantiate classifier with default hyperparameters with kernel=rbf, C=1.0 and gamma=auto
svc=SVC()
# declare parameters for hyperparameter tuning
parameters = [ {'C':[1, 10, 100, 1000], 'kernel':['linear']},
               {'C':[1, 10, 100, 1000], 'kernel':['rbf'], 'gamma':[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]},
               {'C':[1, 10, 100, 1000], 'kernel':['poly'], 'degree': [2,3,4] ,'gamma':[0.01,0.02,0.03,0.04,0.05]}
              ]

grid_search = GridSearchCV(estimator = svc,
                           param_grid = parameters,
                           scoring = 'accuracy',
                           cv = 10,
                           verbose=0)
grid_search.fit(x_tf, y)
# instantiate classifier with default hyperparameters with kernel=rbf, C=1.0 and gamma=auto
svc=SVC()
# declare parameters for hyperparameter tuning
parameters = [ {'C':[1, 10, 100, 1000], 'kernel':['linear']},
               {'C':[1, 10, 100, 1000], 'kernel':['rbf'], 'gamma':[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]},
               {'C':[1, 10, 100, 1000], 'kernel':['poly'], 'degree': [2,3,4] ,'gamma':[0.01,0.02,0.03,0.04,0.05]}
              ]

grid_search = GridSearchCV(estimator = svc,
                           param_grid = parameters,
                           scoring = 'accuracy',
                           cv = 10,
                           verbose=0)
grid_search.fit(x_tf, y)
```

```
clfa = svm.SVC(C=10, kernel='linear')# gamma=0.01
clfa.fit(x_tf, y)
#kfold=KFold(n_splits=5, shuffle=True, random_state=0)
kfold=StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
CVscore=cross_val_score(clfa, x_tf, y, cv=kfold)
y_pred_tf_cv = cross_val_predict(clfa, x_tf, y, cv=kfold)

filename = 'jud_svm.sav'
pickle.dump(bag, open(filename, 'wb'))
ab=CVscore.mean()*100
print('CV accuracy score', ab)
print(classification_report(y_pred_tf_cv, y))
CVscore
```

Figure C.1: Python code of implementation of gridsearch and SVM model

Appendix C2: Implementation of Gridsearch on RF Model with 10 Fold Stratified CV

```
rfc=RandomForestClassifier(random_state=42)
param_grid = {
    'n_estimators': [200, 500],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [4,5,6,7,8],
    'criterion' :['gini', 'entropy']
}
from sklearn.model_selection import GridSearchCV
CV_rfc = GridSearchCV(estimator=rfc, param_grid=param_grid, cv= 10)
CV_rfc.fit(x_tf, y)
```

```
RF_model=RandomForestClassifier(random_state=42, max_features='auto', n_estimators= 200, max_depth=4, criterion='gini');
RF_model.fit(x_tf, y)
kfold=StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
CVscore=cross_val_score(RF_model, x_tf, y, cv=kfold)
y_pred_tf = cross_val_predict(RF_model, x_tf, y, cv=kfold)
filename = 'jud_RF.sav'
pickle.dump(bag, open(filename, 'wb'))
ab=CVscore.mean()*100
print('CV accuracy score', ab)
print(classification_report(y_pred_tf, y))
CVscore
```

Figure C.2: Python code of implementation of gridsearch and RF Model

Appendix C3: Implementation of Gridsearch on NB Model with 10 Fold Stratified CV

```
params = {'alpha': [0.01, 0.1, 0.5, 1.0, 10.0],
         }

nb_grid = GridSearchCV(MultinomialNB(), param_grid=params, n_jobs=-1, cv=10, verbose=5)
nb_grid.fit(x_tf, y)
```

```
BNBclf = MultinomialNB(alpha=0.1, fit_prior=True, class_prior=None)
BNBclf.fit(x_tf, y)
kfold=StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
CVscore=cross_val_score(BNBclf, x_tf, y, cv=kfold)
y_pred_tf = cross_val_predict(BNBclf, x_tf, y, cv=kfold)

ab=CVscore.mean()*100
print('CV accuracy score', ab)
print(classification_report(y_pred_tf, y))
CVscore
```

Figure C.2: Python code of implementation of gridsearch and NB model

Appendix C4: Results of Feature Extractions without Remove Stop-Words

Table C4.1: Feature extractions without remove stop-words of three models

Feature Extraction	***SVM model with 10 Fold Average Accuracy (mean) %***	***RF model with 10 Fold Average Accuracy (mean) %***	***NB model with 10 Fold Average Accuracy (mean) %***
TF-IDF	**91.38**	**93.82**	88.65
BOW	89.44	89.98	**90.29**
Unigram	68.49	70.91	77.36
Bigram	80.59	76.41	79.58
Trigram	88.21	88.72	81.24

Appendix C5: Result of Randomized Search with Stratified 10 Fold Cross-Validations

Table C5.1: Randomized search result with stratified 10 fold CV of judgment model

Feature Extraction	*SVM model with 10 Fold Average Accuracy (mean) %*	*RF model with 10 Fold Average Accuracy (mean) %*	*NB model with 10 Fold Average Accuracy (mean) %*
TF-IDF	**94.83**	**94.92**	91.76
BOW	92.79	93.35	92.84

Table C5.1: Randomized search result with stratified 10 fold CV of penalty model

Feature Extraction	*SVM model with 10 Fold Average Accuracy (mean) %*	*RF model with 10 Fold Average Accuracy (mean) %*	*NB model with 10 Fold Average Accuracy (mean) %*
TF-IDF	**77.85**	**76.27**	66.99
BOW	75.55	75.41	69.37

Appendix D1: Result of Stratified CV Average Accuracy of Three Models with *k* Values (2-10) on Judgment Dataset on Default Parameters

Table D1.1: Average accuracy (%) of SVM model results

Feature extraction	***K-values***								
	2	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
TF-IDF	80.44	85.98	86.68	89.69	90.46	91.77	92.85	93.25	**94.41**
BOW	78.76	82.52	84.67	86.91	87.41	89.96	91.39	92.34	**93.25**

Table D1.1: Average accuracy (%) of NB model results

Feature extraction	***K-values***								
	2	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
TF-IDF	82.45	87.62	86.21	89.81	89.99	89.66	91.69	92.54	**93.02**
BOW	81.66	88.47	86.92	88.67	86.55	90.44	91.51	92.88	**93.78**

Table D1.1: Average accuracy (%) of RF model results

Feature extraction	***K-values***								
	2	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
TF-IDF	84.49	88.25	89.55	92.87	91.46	92.22	93.73	94.88	**96.02**
BOW	82.38	87.96	88.95	90.68	90.94	90.36	91.62	91.33	**93.95**

Appendix D2: Result of Stratified CV Average Accuracy of Three Models with *k* Values (2-10) on Penalty Dataset

Table D1.1: Average accuracy (%) of SVM model results

Feature extraction	***K-values***								
	2	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
TF-IDF	67.27	69.19	70.56	72.89	73.92	74.29	76.68	76.84	**77.98**
BOW	65.25	67.66	68.48	70.92	69.85	70.11	70.53	71.47	**72.66**

Table D1.1: Average accuracy (%) of NB model results

Feature extraction	***K-values***								
	2	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
TF-IDF	53.29	55.23	56.44	58.97	58.69	59.88	60.92	60.42	**61.89**
BOW	62.58	64.14	65.27	68.87	67.41	67.79	68.09	69.18	**70.42**

Table D1.1: Average accuracy (%) of RF model results

Feature extraction	***K-values***								
	2	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
TF-IDF	64.29	67.19	69.44	71.88	71.29	72.98	73.89	74.16	**74.28**
BOW	63.89	65.21	68.86	71.11	70.12	71.48	72.96	73.22	**73.95**

Appendix E1: Human Evaluation Form

Figure C.2: Human evaluation form