DEVELOPMENT OF A PREDICTION MODEL FOR TAX ASSESSMENTS USING DATA MINING AND MACHINE LEARNING TOOLS

BY

Anthony Willa Sampa

2018249240

A RESEARCH PROPOSAL SUBMITTED IN PARTIAL FULFUILMENT OF THE REQUIREMENT OF A DEGREE OF MASTERS' IN COMPUTER SCIENCE

THE UNIVERSITYOF ZAMBIA

COMPUTER SCIENCE

LUSAKA

OCTOBER 2023

COPYRIGHT

DECLARATION

I, Anthony Willa Sampa hereby declare that this dissertation is my original work, and that it has not been submitted in whole or in part for any other degree or qualification. All sources of information have been acknowledged and any direct quotations have been identified as such. I further declare that the research conducted in this dissertation has been carried out in accordance with the ethical principles and guidelines of the University of Zambia, and that all participants in the study have provided informed consent. Any data or materials used in this dissertation that were obtained from third-party sources have been appropriately cited and referenced.

Name: …….

Sign: ……

Date: ……

APPROVAL

This dissertation, by Willa Anthony Sampa has been approved as partial fulfilment of the requirements for the award of the degree of Masters of Science in Computer Science by the University of Zambia.

Examiner 1

Name: …….

Signature: ……..

Date: ……..

Examiner 2

Name: …….

Signature: ……..

Date: ……..

Examiner 3

Name: …….

Signature: ……..

Date: ……..

Chairperson (Board of examiners)

Name: …….

Signature: ……..

Date: ……..

Supervisor

Name: …….

Signature: ……..

Date: ……..

ACKNOWLEDGEMENTS

In this research paper, I would want to sincerely thank everyone who has helped with my effort in this research. Firstly, I would like to thank my research supervisor, Prof Jackson Phiri, for providing me with invaluable guidance, support, and encouragement throughout the project. I am grateful for their unwavering commitment and dedication to our research. I also acknowledge the support of University of Zambia Computer Science for their invaluable input, insightful discussions, and support that have significantly improved the quality of this research paper. Finally, I would want to thank the Zambia Revenue Authority for allowing me to carry out with research and believing in the value it will bring to the institution.

DEDICATION

I dedicate this research to my wife Matanda S. Sampa and our lovely children Seth Willa Sampa and Antoinette Malaika Sampa who have been a source of encouragement and endless support throughout my academic journey. To my parents Alfred and Prisca Sampa, thank you for the support, belief and encouragement which made my research more comfortable. Finally, would like to thank my supervisor and all the lecturers that helped in imparting knowledge and guidance rendered during the research.

ABSTRACT

A nation's economic development still depends heavily on its tax system. The tax administrations in Zambia much like many other countries face many difficulties in the tax collection process, with inadequate compliance being the main one. Therefore, in order to fix the issues and improve total collection, it is critical to be able to identify revenue leakages as much as feasible. In this study, we examined Zambia's tax audit and assessment procedure, which looks for income leaks brought on by fraudulent activity, understated assets, and declaration errors. It is crucial to carefully distinguish cases that are likely to result in smaller, insignificant revenue collection from those that yield significant revenue in comparison with the limited resources used to perform audits and assessments because of the large volumes of audit cases generated by audit selection methods, some of which yield very little collections after the audits. In order to identify the audit and evaluation choices that are most likely to result in a noticeably large revenue collection from these leakages, we created a machine-learning model employing supervised learning. Using the Random Forest technique, we created a prediction model that yielded a 91% ROC curve evaluation. The experiments' outcomes demonstrated that the prediction model created was able to accurately distinguish high value assessment cases for audit from the rest of the cases. This prediction model will help focus the auditing resources in areas that will yield the most revenue and ultimately assist the revenue authority in efficiently raising the nation's tax collections.

Keywords: Compliance, Audit, Assessment, Machine Learning, Tax

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under Curve
COVID	Corona Virus Disease
CRISP-DM	Cross Industry Standard Process for Data Mining
CSS	Cascading Style Sheets
DM	Data Mining
DT	Decision Trees
FK	Foreign Key
FNN	Feed-forward Neural Network
GDP	Gross Domestic Product
GUI	Graphical User Interface
HTML	Hypertext Mark-up Language
JSON	JavaScript Object Notation
LME	London Metal Exchange
LR	Logistic Regression
ML	Machine Learning
OOB	Out of Bag error
PAYE	Pay As You Earn
PK	Primary Key
RDBMS	Relational Database Management System
REST	Representational State Transfer
RF	Random Forest
RNN	Recurrent Neural Network
ROC	Receiver Operator Characteristic
SME	Small and Medium Enterprise
SVM	Support Vector Machine
TAO	Tax Appeals Tribunal

TOT	Turnover Tax
TPIN	Taxpayer Identification Number
UK	United Kingdom
UML	Unified Model Language
URL	Uniform Resource Locator
USA	United States of America
VAT	Value Added Tax
VSC	Visual Studio Code
ZRA	Zambia Revenue Authority

1. INTRODUCTION AND BACKGROUND

1.1 Introduction

A nation's economy is fundamentally influenced by its tax system. Revenue collection is key to the social and economic development of a country. Taxes are collected from income earning citizens and organizations of a country with the view to maintain objectivity in the process. The purpose of the tax system is to collect contributions from residents and businesses in a fair and equitable manner, taking into account the sources of revenue and economic activities of each party. From the standpoint of economics, taxes are a well-considered method of redistributing money from the wealthy to the less fortunate members of society and of funding government spending [1]. The principal concern of a tax authority is to impose as many tax breaks as possible before tackling the difficult issue of collecting additional money to fund public sector operations [2]. The funds raised go toward initiatives for national development, including health care, education, and infrastructure, all of which are meant to enhance the quality of life for the populace and advance the nation as a whole. In Zambia, the tax system is based on self-declaration. Therefore, it's critical to accurately and efficiently evaluate taxpayer statements and determine through assessment the right taxes that people should be required to pay. In general, Zambia's tax collection to GDP ratio is substantially lower than that of affluent nations [3]. The World Bank states that tax receipts that exceed 15% of a nation's GDP are reliable predictors of economic progress [4]. In Zambia some smaller taxpayer exhibit low compliance as they find challenges in the proper management of their book keeping. Other challenges faced are the difficulty in the taxation of cross-border business, smuggling and tax evasion. These challenges are further increased by the high cash economy in the country where audit trails are difficult to maintain and follow [5]. In this study, we created a prediction model for taxpayer assessments using supervised machine learning and data mining to automatically identify tax declarations that result in significant positive assessments. This will increase revenue collection and improve the process' effectiveness and efficiency. The model was developed using the random forest supervised machine-learning algorithm with nine features. The model results were analysed in detail and conclusions drawn on its effectiveness in detecting assessments with significant financial value. We developed a prototype application with two modules to use the model as part of their engine to perform predictions. The first module developed was a distributed system running on web server with a user-friendly interface, which allowed users to interact with prediction model through the application. The other module was a web service API that meant for integration with other external systems.

1.2 Background

Tax audits and assessments lay at the centre of revenue collection and the distribution and admittance of taxes equitably within a country. The historical progression of tax assessments show a dynamic interaction between legislative structures, policy alterations, and public demands. Self-assessment systems increase the efficiency of the tax collection process for tax authorities without having an unacceptable negative effect on the other key characteristics of a well-designed tax system[6]. Tax assessments are a way in which the Zambia revenue authority attempts to estimate the amount of taxes a taxpayer is supposed to pay. With the ever evolving economic landscapes and changing financial patterns, the government faces challenges employing effective mechanisms to produce effective tax assessments. In the assessment community, estimating an institution's net income and capitalizing that revenue stream into use value are conventional procedures [7]. Traditional assessment approaches, which include market comparisons and cost-based methods, now harmonise with modern computer aided tools which help with data analytics and machine learning algorithms in order to keep up with this trend. These high-tech improvements help improve the assessment process but also introduce new forms of complications that need to be considered in modern tax administration. Tax reforms in many cases are required to maintain or restore the delicate balance between fairness and optimised revenue collection. It involves improving trade-offs between revenue generation equity, efficiency and compliance[8]. A very common challenge in the tax assessments process is the disagreements between taxpayers and tax authorities who most of the time have different interpretations of business transaction values and taxable income calculations. In addition to this, other challenges exist as they present several layers of complexities. In Zambia, there is a diversity of income sources which if not handled and analysed correctly present significant challenges in the tax assessment and enforcement process. They come with the risk of divergent interpretations across the stakeholders and therefore may increase disputes and legal battles in the community. This challenge coupled with continued revisions on the general taxes and tax rates by the government in efforts to increase revenue collection adds another layer of complexity. These tax revisions rely on reliable statistical and analytical information in order to make better informed predictions and decisions. In Zambia, the quality of the information is usually poor due to lack of adequate systems to manage information, which leads to poor predictions and ultimately poorly informed decisions on tax reforms and tax rates. The other challenge present is the population literacy levels. Poor literacy levels may be as a result of a wide number of issues such as inadequate

educational infrastructure and lack of initiatives to help educate the general population on tax matters. Taxpayers may feel unfairly treated due to a lack of understanding for how tax systems work and how their tax contributions are used by their governments. Corruption is another very common vice, extending from the citizens to revenue collection officers leading to incorrect and inconsistent assessments in the country. Corruption usually occurs when some taxpayers are willing to pay bribes to tax administration officials who are willing to waive the auditing process of the taxpayers or assess them with a reduced assessment than what it should be[9]. According to a research by Uslaner [10] findings on former Soviet Union nations indicate that when corruption and low-quality public services are present, taxpayers tend to avoid paying their fair share. This affects the general perception and confidence about the equity and fairness of the tax administration system. The revenue authority's human resource to taxpayer ratio is another major challenge in a self-declaration tax administration system. In Zambia similar to other countries generally speaking, very few income tax returns—always much fewer than one percent of all returns—are the subject of a comprehensive tax audit [11]. These challenges emphasise the need for a broad understanding of the complexities involved in the tax assessment process. Tax audits may encourage both compulsory and voluntary compliance[12]. While these many and complex challenges exist and continue to put a strain on the taxpayers and the revenue collection for the country, some indicators demonstrate the existence of additional elements and metrics in the area of tax compliance that may be working well and supporting the existing shortfalls. This may be evident in the fact that, in spite of great expectations, some taxpayers continue to be obedient and completely cooperative from the taxation area. This research looks to propose potential practical and evidence based solutions and improvements to the existing implementation of the tax audit and assessment process in Zambia using computer aided tools for prediction, automation and machine learning. In this research process we aim to take into careful consideration of maintaining fairness and transparency on the tax administration system as we at the same time seek to improve optimize revenue collection.

1.3 Statement of Problem

The Zambia revenue authority aims to optimise revenue collection by increasing revenue collection through increasing revenue streams and improving their efficiency. At the same time they aim to mitigate the threats to revenue collection by increasing compliance levels of the population. Fraud remains one of the biggest challenges to tax compliance among many other challenges. According to the Financial Intelligence Centre FIC trends report of 2021 [13], in

2019 the tax evasion cases by value was recorded at K144,000,000 and by 2021, the value had significantly increased to K722,000,000. Other factors such as corruption, fraud and money laundering continued to maintain elevated values with fluctuations during the same period. In Zambia, taxpayers self-declare their transactions and taxes to the revenue authority. Due to the large number of declarations filed by taxpayers on the tax administration system, it becomes difficult for officers to review all of them and make the necessary adjustments through audits and assessments in order to collect the correct amount of taxes. This challenge coupled with the ever evolving economic landscapes and changing financial patterns governments around the world face challenges employing effective mechanisms to produce effective tax assessments. A study by Ariel [14] suggests that prudent taxpayers are likely to understand that tax authorities' audit intensity is unlikely to reach 100% due to the infrequency of meetings with taxpayers and the lack of resources available to them for a comprehensive audit covering all taxpayers. The assessments are therefore made of a targeted group of taxpayers. Auditors at ZRA based on various conditions initiate audits and assessments to determine whether the tax declarations is correct. At the end of the process, a negative or positive assessment is posted on the taxpayer's account based on the auditor's findings. One of the various ways assessments are created, is via taxpayers' declarations called tax returns. Based on certain checks, an audit is raised and the audit team employs various techniques to come up with the final assessment, which is finally posted on the taxpayer's account. The process auditors go through to generate the assessments is tedious and may lead them to wasting time on cases that may lead to insignificant assessments while the cases that could provide more significant positive assessments may be left out. The current process may also lead to inaccuracies due to the tedious process involved. The deterrence theory of taxation states that increased levels of audits results in higher levels of tax compliance [15]. It is therefore very important to maximise the tax audit and assessment process with the limited resources available in order to increase compliance. In order to automatically identify tax declarations that are likely to result in significant positive assessments, we developed a prediction model for the prediction of taxpayer assessments in this study using data mining and supervised machine learning. This will increase revenue collections and improve the effectiveness and efficiency of the process.

1.4 Aim

To improve tax audit and assessment case selection accuracy using a machine learning model.

1.5 Research Objectives

i. To identify the factors affecting tax compliance
ii. To assess the best performing prediction algorithm for tax assessments
iii. To develop a prototype application that integrates with the prediction model in (ii).

1.6 Research questions

i. What are the major challenges affecting tax compliance?
ii. How can we use data mining and machine learning to identify tax assessments for audit?
iii. Can the enhancement of the current audit process based on model in (ii) help to address the challenges in (i)

1.7 Significance of the study

Among the biggest obstacles that tax authorities encounter is ensuring that the correct amount taxes are collected from the taxpayers in a country. This research will enhance the existing processes by introducing modern and scientific methods to generate cases for audits and assessments. Traditional tax assessment processes tend to be inefficient, time and cost-consuming and laborious. By leveraging on data mining and machine learning algorithms we can automate and improve the assessment process and reduce errors and inaccuracies. This research will also help to identify unusual patterns in the declarations data which would indicate fraudulent behaviour from taxpayers. The research proposes an automated process for the detection of high value assessment cases for audit thereby improving the efficiency of the human resources.

1.8 Scope of the study

This research will use tax assessment data from the Zambia revenue authority to develop a prediction model using data mining and machine learning. We will develop three models using different algorithms, compare their performance and select the best performing model. We will further develop an application prototype with two modules that will demonstrate how

prediction model will be integrated with the other systems and how users will interact with it in order to make predictions.

1.9 Organisation of the Dissertation

There are five chapters in this dissertation. The introduction and background information for the research are covered in Chapter 1. It will contain the dissertation's goals, objectives, and research questions in addition to introducing the problem statement. The chapter will also give the significance of the study and finally the scope in which the research is to be conducted. Chapter two will highlight the literature review from different literature in relation to the research. We will review the different taxes in Zambia, the challenges and influences in tax compliance and will review the audit process and identify the gaps in the process. We will finally review different machine learning algorithms and review other related research conducted. In chapter three, we will highlight the methodology that is used in the research. It will discuss the framework that will be used in the process and methods of data collection, understanding, analysis, design and implementation. The chapter will also highlight the development of the models and application and show the different tests that will validate the developments. Chapter four evaluates and discusses the results obtained from the previous chapter. The chapter further analyses the results and compares and presents them. Chapter five will discuss the solutions to the research questions highlighted in chapter one. It will present the conclusion of the research and provide recommendations based on the finding of the research study.

1.10 Chapter Summary

Chapter one provided an introduction the research. The chapter provided the background and the problem statement of the research study. It further outlined the aims, objectives and research questions. The chapter presented the significance and scope of the research. Finally it outlined the organisation of the dissertation.

2. LITERATURE REVIEW

2.1 Introduction

In this section, we explored the taxation system in Zambia, factors affecting tax compliance and past research that was done on similar kinds of problem. We reviewed how the machine learning approaches that have been used in the past to resolve similar business problems across the different industries. We learnt how supervised machine learning helped in the development and implementation of an accurate prediction model to address the problems. We also reviewed the specific techniques that have been used in order to have an understanding as to how they relate to our research project. This gave us an insight into which supervised learning techniques would be more suitable to our problem in order for us to yield the best results.

We also reviewed research works that have been done that look at the factors that may lead to taxpayer's returns getting a positive or negative assessment after an audit is done. This review helped us with coming up with a solution that is effective and relevant help address the problem discussed in the problem statement.

2.2 Taxes in Zambia

The primary purpose of tax collection in a country is to collect and provide funds to the government for government expenditure programs. A tax can be operated as a tool to restructure revenue in a country's economy to help decrease inequality or can be used as a mechanism for guideline to encourage or discourage specific actions in order to enhance community wellbeing [16]. The wealth of any nation is assessed by its performance in infrastructure provision through its construction industry. The construction industry is large, unstable, and requires tremendous capital expenditures [17]. In the general context, the objective of taxation is similar all over the world. The solution to a good tax administration system however is most of the time specific to a country. A country's policy influences how the taxes and tax administration systems are structured. The country's strengths are articulated in its capability to make the revenue collection equitable, fair and beneficial to the majority of the country's citizens. In developing countries such as Zambia, there is a high appetite for government expenditure due to the many social and economic areas that require attention. In developing countries, the challenges faced by citizens are common. The common challenges in such economies include poor health facilities that cannot satisfy the population, poor school systems, infrastructure and poor roads required to move essential goods and services effectively. According to Kaliba, Muya and Mumba [17], it was found that in developing

countries, road development takes up a large component of the construction industry. This is interpreted that a substantial segment of the national budget on infrastructure development is channelled towards road development projects. It is for this reason that certain tax models can be shared and can work effectively in countries with similar economic positions without a lot of customisation. In developing countries, there is a high dependency on tax collections in order for the economy to thrive. Revenue authorities are constantly seeking to ensure they are collecting the right amount of taxes by identifying, fixing revenue leakages and increasing compliance levels of the tax population.

2.2.1 Types of taxes in Zambia

In Zambia, the law states that any individual or company earning an income is required to contribute to the country's tax collection. Taxpayers in Zambia are engaged in many various income-generating activities. Some are high earning, some are medium while others are income earning. It therefore it would be unfair to collect the same amount of tax from all taxpayers due to the many variations in the business such as business type, income generated and many others. Collection of taxes from a narrow base often entails very high tax rates. This usually occurs where it is difficult to track some taxpayer normally in the informal sector. According to Easterly [18], it is said that in Argentina, more than 80 percent of income goes undeclared. In Ivory Coast, a country where there is reliance on formal sector tax, employment in the private formal sector accounts for only 1.4 percent of the populace. Repeated attempts to increase tax collection from the formal sector in Ivory Coast have been met with failure, as there is extensive taxes evasion, with an average effective tax rate of about 48 percent [18]. In order to make the tax collection process equitable for businesses and individuals earning income, Zambia has a number of different tax types and these entities will fall into one or more tax types depending on the types of income generating activity they are engaged in and the amount of income they generate. The crucial attribute of the income distribution problem is low rural and high urban income levels, and that actual incomes are affected by the rural-urban terms of trade [19]. The government comes up the tax types that are applicable in the country including the rates and other policies. These are normally reviewed throughout each year and may be revised in the annual national budget. Some of the policies may be designed to encourage certain industries while some policy changes may be meant to give some sort of relief to taxpayers under certain conditions. An example of this is if the government wanted to discourage a certain popular product that may be destructive to the health of the citizens such as cigarettes, it would introduce a new tax type specifically for it or increase the rates already existing that affect that

product. Miskam, Noor, Omar and Aziz [20] in their research that dealt with the determinants of tax evasion on imported vehicles, noted that by presenting certain taxes called “sin tax” to cigarettes, it would be expected that their usage would be reduced to encourage a healthier way of life and protect the environment from damage caused by cigarette smoke. Another example is if the government wanted to encourage a cleaner and greener environment, they would decide to increase taxes on production or importation of plastic bags. The other similar example of this is during the COVID 19 pandemic where many businesses suffered due to the closures of borders, airports, shops, bars and restaurants. In such situations, the government may decide to provide tax relief in the most affected industries in order to stimulate production and recovery. On the other hand, while other industries suffered during the pandemic, for others an opportunity presented itself and they were able to increase their earning. An example of this is the telecom industry, which received a very high demand for their goods services due to many companies allowing their employees to work remotely. According to a study by Farzami, Gregory-Allen, Molchanov and Sehrish [21] which was comparing pre-COVID with COVID services, it was discovered that telecommunication services is the only industry that experience an increase in liquidity. Due to the restrictions and social distancing, companies encouraged their employees to work away from their offices while their main offices remained closed. In a situation like this, the government may decide to increase the tax rates or introduce a new tax type for entities in that industry so that the collections can cushion the negative effects of the pandemic in the economy. In Zambia, campaigns during elections have also had an effect the tax policies of the governments [22].

2.2.1.1 Pay As You Earn (PAYE)

The pay-as-you-earn (PAYE) system introduced in the year 1966 as of type of withholding tax from employment income [23]. The Zambia Revenue Authority defines Pay as You Earn as the process of withholding taxes from employees' total income in accordance with their wages [24]. Earnings from employment may include salaries and wages, overtime and bonuses, gratuities and allowances, cash benefits and commissions [16]. With this system, the employer has the authority to determine the amount of tax that each employee must pay, deduct the tax from their wages, and send the money that has been withheld to ZRA [25]. The amount of tax which the employer deducts from any pay depends on the employee’s total gross pay and the current applicable tax rates in the period. The PAYE rates are subject to change according to the annual national budget. According to the ZRA annual report [26], 2020 saw excess collections of K4,135.1 million, or 13.5 percent, above the K30,628.3 million yearly objective

reported by the Direct Taxes Division. Strong performance in PAYE, mining company tax, withholding taxes, and mineral royalty tax—which reported respective surpluses of K1, 897.7 million, K1, 614.4 million, and K117.1 million—was the reason for the positive performance in direct taxes. Pay as you earn was the highest contributor of all the tax types with a total value of 14,229.2 million [27]. The value of tax audit assessments in 2020 was recorded at K135.92 million [26]. An employee may apply for a tax refund from ZRA if they leave employment are currently unemployed. PAYE refund errors may arise from pay-roll errors or unemployment repayments which are because tax tables were misused or not used at all. Declarations and payments for PAYE are due every 10th of every month.

2.2.1.2 Turnover tax (TOT)

The tax known as Turnover Tax (TOT) is imposed on the gross sales and turnover. These are proceeds, yield, takings, income, revenue, and earnings. Any company operating with yearly sales of K800,000.00 or less is qualified to register for turnover tax. [25]. The current rate for turnover tax is 4 percent but is subject to change with annual national budget reviews. Turnover tax is a simple tax meant for the lower income earning taxpayers. In the year 2020 the value of tax audit assessments was recorded at K10.37 million [26]. Declarations and payments for turnover tax are due every 14th of every month.

2.2.1.3 Income Tax

The Income Tax Act, found in Chapter 323 of the Zambian Laws, governs income taxation in Zambia. Income tax is a type of tax that is levied on the earnings of partnerships, limited corporations, self-employed persons, and employees' emoluments. Income tax must be paid on profits by all individuals who generate profits. [25]. Income tax applies to entities that have a turnover above K800, 000.00. This implies that an organization cannot simultaneously register for turnover tax and income tax. The Income tax act allows for deduction of certain items of expenditure when determining the taxable business profits which reduces the taxpayer's overall liability. Deductions may be made for expenses incurred wholly and solely for the benefit of the business, revenue rather than capital expenditures, losses carried forward from the same source under specific circumstances, donations to recognized public benefit organizations, and other circumstances stipulated by Zambian law. In 2020 income tax had an audit assessment value of K1,724.55 million [26]. The income tax category of taxpayers are obliged to submit in their provisional return before 31st March of the charge year with an estimated liability. They

are also required to make provisional payments towards that liability and file the final return accompanied by financial statements the following year before 21st June.

2.2.1.4 Withholding Tax

Withholding tax is a mechanism that allows for a purchaser of services and goods to withhold the tax due as they are paying the seller. The amount withheld is the tax that the seller was meant to remit to ZRA after the transactions is completed if the withholding mechanism is not in place. The purchaser submits the tax amount to ZRA on behalf of the seller. A number of different types of payments will have withholding tax under Zambian law such as dividends, interest and royalties, commissions and public entertainment fees. Withholding tax recorded K54.76 million [26] value of audit assessments in 2020. Every month on the 14th, withholding tax returns and payments are due; the tax point is when the income is paid.

2.2.1.5 Property transfer tax

According to Zambian law, property transfer tax is a levy levied on the transfer of property from one entity to another under the Property Transfer Tax Act, Cap 340. The property may be sold to any other entity, transferred within a group of firms, or transferred by an individual to a member of their immediate family. Depending on the form of transfer, the tax rate may change. The seller submits the return with the details of the property, the buyer's details including the value of the property to ZRA and an assessment is done. The actual value is determined and an assessment notice is sent to the taxpayer then they make the payment towards the assessment. Unlike acquisition-value property tax, which is based on the property's worth at the time of purchase, conventional property tax is based on the property's current market value [28].

2.2.1.6 Mineral Royalty

Comprehending the mining process and how it cannot always translate into increased revenue for the mining house is the first step towards comprehending the principle of royalty. It is crucial to understand that a mineral ore only gains value after the mineral is extracted and transformed into a product that can be sold [29]. The mining company only realises profit and value after all expenses and costs have paid off which may be after a long time. One of the earliest and most successful theories of mining taxes is the theory of royalties. The agricultural rent concept establishes that productive land is more valuable than unproductive land [29]. The money paid to the government for taking minerals out of the earth is known as a mineral royalty

[16]. The norm value, which is the monthly average London Metal Exchange (LME) Cash price per metric ton multiplied by the quantity of the metal or recoverable metal traded, is the basis for mineral royalty on base and precious metals [16]. In 2020 MR tax had an audit assessment value of K32.51 million [26]. Mineral royalty taxpayers must submit their returns by the 14th of the month following the month in which the minerals were sold, at the latest.. The table below shows the current rates for mineral royalty as at January 2022.

Table 1: Mineral royalty tax rates [25]

Description	Mineral Royalty Rate
Base Metals (Other than Copper)	5% on norm value
Energy and Industrial Minerals	5% on gross value
Gemstones	6% on gross value
Precious Metals	6% on norm value

2.2.1.7 Presumptive tax

When it comes to dealing with income or activities that are challenging to tax, such the informal sector, presumed taxation focuses on estimations of tax payable [25]. Thuronyi [30] defines "hard to tax" as those entities for which applying the regular income tax or VAT regulations would be nearly difficult given the administrative resources required for this purpose. Presumptive tax aims to broaden the tax base despite the difficulty in enforcement present in the informal sector such as bus and taxi operators by simplifying the collection processes. By its nature, presumptive tax revenue from the individual taxpayers is not easy. Bird and Wallace [31] state in their research that when the cost of such activities exceeds the benefits, presumably the process should be stopped. Presumptive tax rates are often lower than ordinary tax rates, and the tax is computed as a fixed proportion of the taxpayer's gross receipts or turnover. This simplifies the tax calculation process and reduces the compliance burden for taxpayers. In Zambia, under Presumptive Tax, only people and partnerships are subject to this tax.

2.2.1.8 Base tax

Another sort of presumptive tax that is applied to individuals is base tax, which is K365 annually and is levied in cases when the Zambia Revenue Authority is unable to determine the taxpayer's income. The income tax act's section 64 specifies the tax. Those who don't know how much money they make but do receive some sort of income are responsible for paying

this tax. Nhekairo [16] describes base tax as a hard-to-measure levy on marketers and small enterprises. Base tax results from an estimated assessment; nevertheless, ZRA reserves the right to make additional assessments in the event that information regarding an individual's actual or nearly actual tax due becomes available. Since the Commissioner-General is estimating tax as required by the Income Tax Act, no declarations need to be completed [25]. There is no need to maintain any books of accounts for tax reasons when base tax is applied. Because it costs more to enter the formal economy in developing nations than in wealthy ones, there are more informal sectors there than in affluent nations [3].

2.2.1.9 Gaming and betting Presumptive Tax

Gaming and betting Presumptive tax is a tax introduced on persons engaged in activities in gaming and betting as their income and will therefore not be eligible for income tax or turnover tax. The table below shows the rates on the types of games as at January 2022.

Table 2: Gaming and betting tax rates [25]

Type of game		**Monthly Tax Rate or Monthly Tax Amount**
Live Casino games		gross takings of 20 %
Machine Casino Games		gross takings of 35 %
Winnings from Lottery		gross takings of 35 %
Betting		gross takings of 25 %
Gaming:	Slot Machines (Bonanza)	K250 /machine
	Gaming Machines (Limited Pay Out)	K500 /machine

2.2.1.10 Value added tax (VAT)

The value-added kind of consumption tax on commodities that was imposed at the production stage was first introduced in France in 1954 and is known as value added tax (VAT) [32]. Every country using VAT sets its own unique rules on the administering the tax but the basic principle remains the same. VAT is a consumption-based tax that is imposed across the supply chain at every stage when value is added to goods or services, according to the Zambia Revenue Authority. Since VAT is dependent on consumption, the main ways to legally avoid it are to consume solely zero-rated or exempt supply or to avoid consuming any standard-rated goods

or services [25]. According to Keen and Lockwood [33] Over 130 nations impose value-added taxation (VAT), which typically generates 20% or more of total tax income. It is extensively used in sub-Saharan Africa and other regions, where it has served as the cornerstone of tax reform in numerous developing nations. By many measures, the most important change to tax administration and policy in recent decades has been the introduction of the Value Added Tax (VAT) [33]. Typically, businesses are required to register for VAT, which entitles them to collect tax on sales and allows them to recover tax on inputs if a certain threshold is reached [34]. Value Added Tax is paid in Zambia by the last party in the supply chain who is not registered for VAT. Individuals who are registered for VAT must pay to the Zambia Revenue Authority the output VAT that exceeds their input VAT, as well as claim back any input VAT they incurred over the course of their business through the return [25]. Standard-rated goods are subject to a standard VAT rate of 16%; zero-rated goods are subject to a VAT rate of 0%; and exempt goods are not subject to any VAT. In 2020 value added tax had an audit assessment value of K1,674.84 million [26]. Declarations and payments for value added tax are due every 18th of every month.

2.2.1.11 Excise duty

Excise taxes can be seen as taxes on production even though they are typically categorized as taxes on consumption [35]. Excise duty is a levy levied at any point along the production or distribution process on specific items or products, regardless of whether they are produced domestically or imported. It is calculated based on the products' weight, strength, or quantity as well as their worth [25]. Every manufacturer of an excisable product in the Zambia is required to be registered and licensed with the Zambia Revenue Authority except for oil marketing companies who are licenced by the energy regulation board, mobile network operators and internet service providers licenced by Zambia information and communications authority and importers and distributors of cigarettes [25]. Therefore any entity manufacturing, distilling, mixing or brewing all types of spirits, wines, any transportable beverage with a volume percentage of greater than two percent alcohol, opaque beer, cigarettes and other tobacco products, electrical energy, cosmetics, plastic Carrier bags, fuel oils and gases, hydrocarbon oils such as petrol, diesel need to be registered for excise duty. The licence is annual and the taxpayer is expected to apply for renewal before 31st September of every year. Declarations and payments for value added tax are due every 15th of every month. In 2020 excise duty had an audit assessment value of K179.47 million [26]. The table below shows the summary of the audit assessments by tax type of the year 2020.

Table 3: Audit assessments by tax type 2020

Tax type	Audit assessment (K million)
Pay As You Earn	135.92
Turnover tax	10.37
Income Tax	1,724.55
Withholding Tax	54.76
Mineral Royalty	32.51
Value added tax	1,674.84
Excise duty	179.47

This research was conducted on and focussed on the value added tax VAT tax type. Since all the tax types are self-declarations and are all subject to audits and assessments, they have the same general function and objective. The prediction model developed could be extended to all other tax types with minor modifications.

2.2.2 Tax audits

The Zambia Revenue Authority receives self-declarations from taxpayers regarding their business activities. The client generally pick one of two actions when declaring to the revenue authority: (1) disclose all income and pay the applicable taxes right away; or (2) report a portion of their actual income but withhold some of it. If they opt for the latter, their reward will be contingent on both the possibility of being audited and the audit's efficacy, should it be carried out [36]. When tax reporting is optional, as it is in the United States income tax system, unauthorized audits are the primary means of enforcing the tax rules. If it is found that the taxpayer underreported their taxable income, penalties are frequently imposed [37]. It's possible that taxpayers do not always intentionally declare lower income than they actually make. It can sometimes be the consequence of inadequate or inaccurate accounting data, and other times it could be the result of the taxpayer's ignorance of the tax laws. The taxpayer may in certain circumstances overstate their income and pay more in taxes than they actually need to, in which case they would be entitled to a refund of taxes. Examining the veracity of the claimed or assessed tax is the primary goal of tax audits [25]. An activity or series of actions carried out by a tax inspector to ascertain whether a taxpayer is in accordance with applicable tax rules and regulations, including reviewing the taxpayer's tax declaration, is known as a tax audit. This comprises checking the business financial records of an entity to make sure the taxpayer provided accurate information in the tax declaration [25]. Tax audits have many

benefits to both the tax authority and taxpayer in helping them run their businesses respectively. It facilitates for tax education in the process and may provide suggestions for improvement in handling tax matters to the taxpayer under audit. It assists taxpayers in determining the gaps in their accounting system. Additionally, audits support the enhancement of the numbers' dependability and trustworthiness when they are presented to potential buyers. ZRA is legally permitted to audit any kind of business or individual return for a period of six years following the year's end. If fraud is suspected, the investigation term does not have to be six years; it can extend back as far as the fraud's inception, which may be longer than the specified time frame [25].

2.2.2.1 Audit process

Tax audits are mainly prompted by two actions. The first is that audits can be conducted at any time on an unlimited selection of taxpayers for compliance purposes. The second reason for a tax audit is the possibility of noncompliance with tax regulations. When a tax inspector or auditor notifies a taxpayer of a mistake that needs to be corrected, or requests more information about any item(s) on a declaration they filed, an audit is started. The notice includes the period, the tax type(s), the nature of the audit, and the books and records that are anticipated to be examined [25]. It is anticipated that the taxpayer would reply to the notification in a timely manner. The auditor then gets in touch with the taxpayer to schedule a convenient time to meet, either at the taxpayer's home or, if it's a business, at a ZRA office. The auditor will go over the kinds of records that a taxpayer must submit as well as the intended audit methodology and processes during the meeting. During the audit, the taxpayer is expected to cooperate with auditors throughout and provide honest and accurate information at all times. At the end of the audit, the auditor submits their findings to the authority and posts an assessment on the taxpayer’s account at ZRA and the taxpayer is notified of the assessment posted.

2.2.2.2 Types of audits

There are processes in ZRA that direct the audit case selection process. The process of choosing audit cases adheres to the criteria that ZRA specified. The breadth of an audit and the degree of information it covers can vary.

a) **Thorough examination (both integrated and single tax types audit)**
   This is a thorough audit that examines many risk categories. It may concentrate on a single tax type or make cross-taxation cuts (integrated). This kind of audit

examines the taxpayer's entire financial situation over a minimum of one charge year [25].

b) Issue audits

This audit examines a certain problem (region or item) within a specified time frame (one declaration period, i.e., one month, one year, etc., depending on the situation). These audits can be added as suggested cases and are started using credibility criteria [25].

c) Credibility audit

This audit's purpose is to verify the veracity of a particular declaration that might have fallen short of certain requirements [25].

d) Refund audit

This is a reference to audits that are brought about by refund claims [25].

e) De-reg audit

This is the term for audits that arise when a taxpayer closes their doors or deregisters from a particular tax category [25].

f) Educational audits

These audits are meant to enlighten the taxpayer on their rights and obligations as well as provide guidance on particular matters [25].

2.2.2.3 Post audit

After conducting an audit, the auditor notifies the taxpayer or their representative of any adjustments made before concluding the audit. During the audit's reconciliation stage, the taxpayer may bring up any information they may have that the auditor may not have taken into account or if they have cause to think the auditor may have erred. Following reconciliation, an audit agreement that expressly states the areas to which the taxpayer has agreed or opposed must be signed by the auditor, the taxpayer, or his designated representative [25]. An audit's output is an evaluation of the tax and, if applicable, liabilities that the auditor records on the taxpayer's account. ZRA may, using the information at its disposal, determine the tax due and interest owed on any income that has been omitted; alternatively, it may increase an estimated assessment and impose additional penalties for noncompliance with particular tax laws or rules as prescribed by the various tax Acts. [25]. If there is an assessment, it must be paid within 30 days. In accordance with the tax acts, the taxpayer is required to pay for their assessment or request a refund, based on the audit's findings. If the taxpayer is still not satisfied with the objection decision, they have the option to appeal to the Tax Appeals Tribunal (TAO) and, if

they so choose, the Supreme Court for a final decision. The client has the right to begin the objection process for the items that they have not agreed to within 30 days [25]. The number of times a taxpayer may be audited is unlimited.

2.2.3 Tax compliance

Compliance is one of the essential factors for a successful taxation system. Femidah and Phiri [38] defined tax compliance as following or being in compliance with laws or regulations. James and Alley [39] defined tax non-compliance as the inability of the taxpayer, whether via deliberate or inadvertent means, to accommodate their tax obligations. A taxpayer who complies with national tax regulations is one who has fulfilled all of their tax obligations. This involves making certain that the tax authorities receives the necessary tax declarations by the deadline. In accordance with the law, taxpayers must also make all required contributions toward their obligations on schedule. To fulfil their responsibility, tax authorities seek to enhance tax populations' compliance levels. They are concerned with the various factors that lead to low tax compliance and attempt to put in measures to mitigate them. The tax declaration system used in Zambia is a self-declaration system. In a self-assessment taxation system, the taxpayer calculates how much tax they owe from their earnings and makes their declaration for the tax period due to the tax authority. When the self-assessment declaration is made, the tax authority may decide to agree with it or disagree and issue an assessment based on their assessment and allow the taxpayer to respond to it. Self-assessment tax declaration systems are common in many countries around the world. This process makes it easier and manageable when dealing with a large number of taxpayers. The aim of these assessments is ensure that as much as possible the right taxes are paid to the government this may mean that due to the taxpayer’s errors in calculations and understanding, they may have a liability with the authority or they are in a refund position. The overall objective of tax authorities is to increase taxpayer compliance which consequently leads to increase in revenue collection. According the ZRA tax statistics of 2022, the trend from 2021 to 2022 showed that in 2022 the authority failed to meet the revenue collection targets compared to 2021 when the target was exceeded. The authority also failed to meet its target in the year 2023. The table and chart below shows the revenue against the target trend from 2021 to 2023.

Table 4: Revenue Vs Target (K'millions)

	2021	2022	2023
Target	59,076	90,740	103,100
Revenue	83,573	89,937	100,600

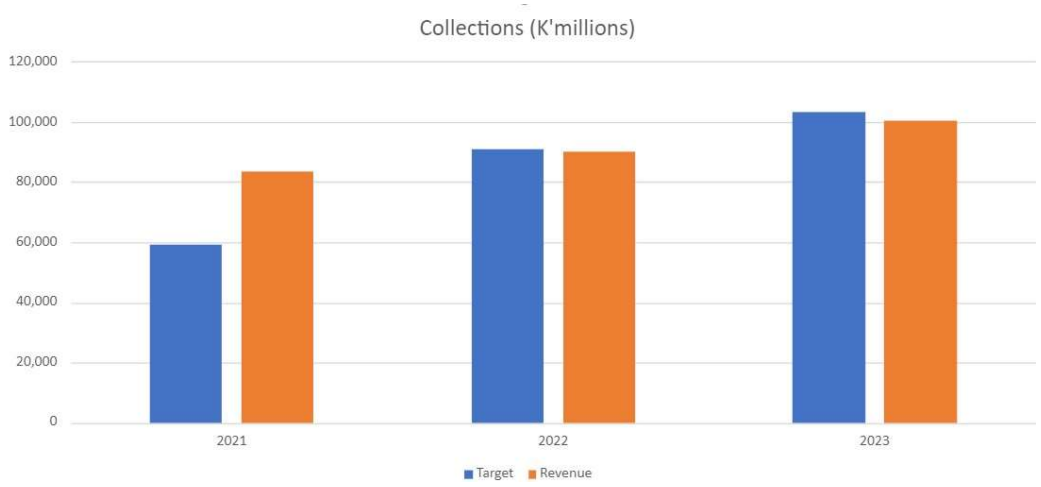


Figure 1: Revenue Vs Target (K'millions)

This could have come about by many different factors including inadequate resources, new players on the tax market, tax fraud and under performance in the mining sector. From an analytical perspective, these steep variations need to be reviewed critically as they could expose challenges that were previously unknown within the taxation space.

2.2.3.1 Influences on tax compliance

It is generally accepted that an increase in the formality of taxpayers is an integral part of the process in which tax collection increases [40]. Atawodi and Ojeka conducted research on the variables influencing tax compliance among small and medium-sized businesses (SMEs) in north central Nigeria [41]. In their research, they highlight that generally speaking, SMEs were defined by evolution, inventiveness, and uncertainty. It would be necessary to have a solid understanding of SMEs' features. The fact that the SME sector was diverse and ranged in size from tiny retail stores to highly compensated professionals and significant manufacturing firms is one of its distinguishing characteristics. The organizational structure of SMEs is also likely to differ from that of professions, partnerships, small corporations (public or private), and sole

proprietorships (with or without workers). Ultimately, they discovered that the two main things contributing to SMEs' non-compliance with tax laws are high tax rates and complicated filing processes. Only slightly affected by other factors, such as multiple taxes and inadequate illumination, were the SMEs surveyed tax compliance rates.

In many instances, non-compliance from taxpayers comes from ignorance and misconceptions within the general population. Some taxpayers feel are there tax contributions are not put into good use by the government. Others feel the tax rates are too high in proportion to their income. Some, however, choose to continue not complying since they believe the procedure is too difficult for them to comprehend. This lack of proper knowledge may actually lead to the taxpayer paying more that they should have initially been paying due to penalties and interest, which could be easily avoided if adequate taxpayer education is provided.

Kornhauser [42] carried out a research on tax morale which she described as the referring to taxpayer attitudes and beliefs. In her research, she explored the connection between taxpayer's beliefs and behaviours. She noted that a taxpayer's belief that someone else is cheating may have an impact on their own standards, depress tax compliance, and alter their compliance behaviour. A key factor in the creation and influence of norms is group identification. A person is more likely to internalize a group's norms and cooperate—that is, obey them—the more they identify with the group. Stated differently, cooperation stems from the need to establish and preserve a positive identity [43]. However according to Raskolnikov [44], a person may adhere to a group's rules even though they do not share their identity, if only for logical reasons like reputation. Adherence to legal requirements typically suggests a person's dependability, honesty, or trustworthiness. According to Kornhauser, internal incentives to abide with tax regulations may also be favourably influenced by affiliation with a group smaller than the country. Business-oriented individuals might accept it as the standard if, for example, corporate leaders emphasized the significance of paying taxes on both a personal and corporate level. Similar to this, having a revered figure like a local celebrity or a prominent member of a group like a minister stress the need of tax compliance could increase compliance.

2.2.3.2 Tax evasion

Payment of a suppressed amount or failure to pay any amount from one's taxable income that is required by law is considered tax evasion. Typically, evasion occurs when a taxpayer knowingly chooses to underreport their income or submit a disproportionately large refund claim [45]. According to the International Monetary Fund (IMF) (2015), in 1990, only 60% of

countries worldwide had a tax ratio of 10%, and by 2013, that percentage had increased to 75% of all countries [46]. The low compliance rates are intimately linked to the act of tax evasion. Wallschutzky [45] discovered from studying two categories of people who abscond from the mainstream public. Based on the answers he obtained from the tax evaders, it was discovered that one of the main causes of tax evasion is the belief that taxes are excessive and do not provide value for the taxpayer. Additionally, he discovered that the tax evaders believed that the government did not use taxpayer funds wisely and that low-income earners bore a disproportionate share of the tax burden.

If most taxpayers anticipate that many others will follow suit, then most will avoid paying taxes. Conversely, if most taxpayers believe that few others will evade, then most will pay taxes. Both individual and corporate taxpayers engage in these behaviors, despite the fact that, on the whole, businesses are more prone than individuals to evade taxes [47]. In contrast to a higher value for tax revenues needed for public goods and services, which has a favorable influence on tax compliance, excessive uncertainty and sunk costs of tax enforcements have a negative effect on tax compliance, according to study by Bruno[47] on tax enforcement, compliance, and morale. Additionally, he discovered that people who use third-party reporting typically avoid taxes at a lower rate than people who self-declare.

Evasion is committed in a variety of methods with the intention of avoiding detection by the tax administration. While some schemes are highly complex and may include numerous valid procedures both inside the system and outside of it, others tend to be relatively simple and clear. One such method involves a number of separate activities and procedures that, taken separately, do not constitute fraud but, when combined, demonstrate that the taxpayer's goal was evasion [48].

The trends report published by the Financial Intelligence Centre in 2021 [13] showed that in 2021 there were 17 reports of tax evasion cases amounting to 722 million kwacha which was an increase in value from 717 million in the year 2020 and 144 million in the previous year. The table below shows an extract from the report.

Table 5: Disseminated Reports by value and number

Suspected offence	2021		2020		2019	
	No of reports	ZMW (Millions)	No of reports	ZMW (Millions)	No of reports	ZMW (Millions)
Tax Evasion	17	722	24	717	17	144
Corruption	4	1,276	14	2,228	4	332
Fraud	3	20	6	26	8	53
Money Laundering	16	1,543	3	4	6	450
Terrorist Financing	4	0.33	1	0.16	1	2
Others	0	0	13	166.84	8	3
Total	44	3,560	61	3,142	44	985

According to the Zambia Revenue authority 2022 annual report [49], in the year 2022 the authority prosecuted 31 cases which was an increase from 21 cases in the previous year. 16 cases were concluded out of which 12 were convictions, were acquittals while one case was withdrawn. By the end of the year 2022, 18 cases were under prosecution while one was at planning stage. The table below shows a summary of these cases.

Table 6: Prosecuted cases in 2022

Cases		Customs	Inland Taxes	Penal Code-Chapter 87 of the Laws of Zambia	Total
Brought forward		13	2	1	16
New cases		3	9	3	15
Total		16	11	4	31
Concluded	Conviction	11	-	1	12
	Acquittal	2	1	-	3
	Withdrawn	1	-	-	1
	Referred	-	-	-	-

According to the Zambia Revenue authority 2022 annual report [49], the authority had 48 civil cases under litigation. 13 were before the Tax Appeals Tribunal, 9 before the Industrial Relations Court, 3 before the Subordinate Court, and 2 before the Supreme Court, 2 before the Court of Appeal, and 1 before the High Court. The table below summarises the cases and shows the trend between 2021 and 2022.

Table 7: Civil litigation cases 2021 and 2022

Type of Court	2022	2021	Variance
Supreme Court	2	6	-4
Court of Appeals	2	0	2
High Court	19	12	7
Tax Appeals Tribunal	13	19	-6
Industrial Relations Court	9	10	-1
Magistrates Court	3	1	2
Total cases	48	48	0

In the year 2022, the authority received a total of 65 tax appeals related to assessments while in 2021, it received a total of 68 tax appeals related to assessments.

2.3 Machine learning models

In this research, we used data mining and machine learning to create a prediction model for tax assessments. Machine learning is a technique used to 'teach' a computer how to react to future activities that it may encounter without explicitly programming it on how to respond to specific input. This gives the computer a level of artificial intelligence (AI), which allows it to be able to respond to real world problem similar to the way a human being would. Mahesha [50] describes the main purpose of machine learning as to be able to learn from data. Large data sets are subjected to statistical algorithms in machine learning in order to identify trends and make choices. A trained model is the end result of the machine learning process, and it may be applied to new data to make judgments. The following diagram illustrates how computing complexity and sophistication are distributed over a continuum.

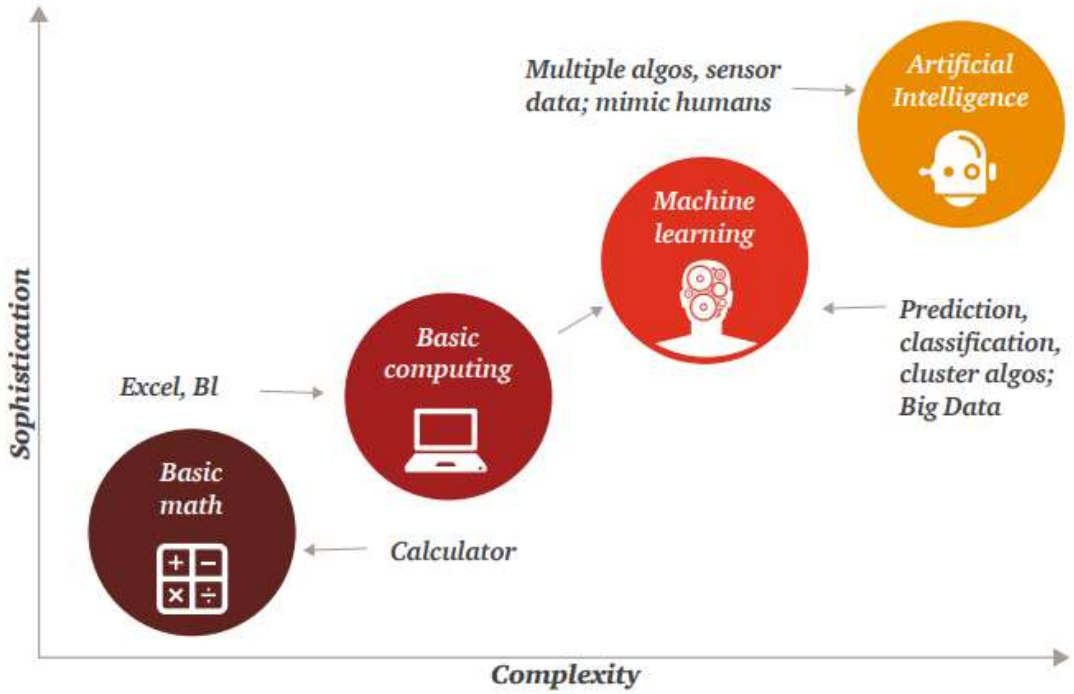


Figure 2: Sophistication and complexity in a continuum [38]

When testing a model with test data, the general goal of a learning process is to identify a function that minimizes the risk of prediction error, which is expressed as the difference between the generated output and actual results. [51]. Various algorithms and techniques are available to use to come up with a usable model. The use of these algorithms varies depending on the problem at hand. It is important for a data analyst to understand which one to use and when to use it.

2.3.1 Supervised learning

One kind of machine learning technique is supervised learning, which is applied when the model is trained with known training data and known outputs from that input. Using supervised learning, an algorithm or function is trained to calculate output variables from supplied data that contains both input and output variables. [51]. Some of the common supervised learning techniques are Logistic regression, K Nearest neighbour, Naïve bayes, Support Vector Machine (SVM), Artificial Neural Networks (ANN), Decision Trees (DT) and Random Forest (RF).

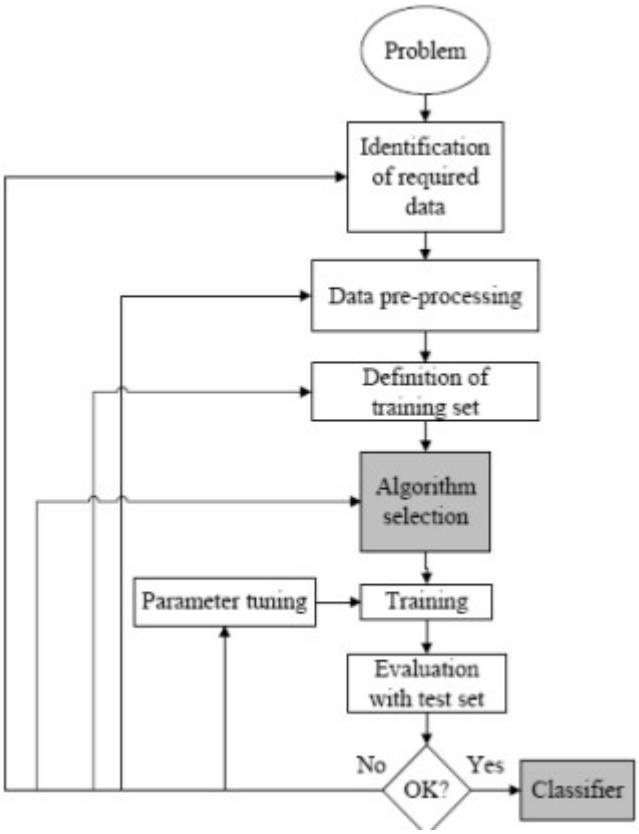


Figure 3: Supervised Learning process [40]

In the research, we review and evaluated literature on supervised learning algorithms that have been used in the past to solve similar problems. From the evaluations, we selected three algorithms that most suited algorithm on which to build our model.

2.3.2 Classification

One method in supervised learning is classification, which involves grouping or categorizing the data into groups in order to learn about it. A classifier takes a database of records, each with a class label. It then creates a clear and meaningful description for each class, which may be applied to the classification of records that come after it [52]. When new data is fed into the model that is created using this technique, the data expected to be categorised into the groups according to the learning. This type of analysis can help us to understand large amounts of data and draw some conclusions from it. Classification can be used to process a wider range of data since it anticipates categorical (discrete, unordered) data [53]. Dichotomous classification is one kind of classification that is applied in classification. The classifier creation method receives a training set of records, each of which has a class label attached to it. A set of attribute values defines each record. Attributes with discrete domains are referred to as categorical,

while those with ordered domains are referred to as numeric [52]. In this type of classification, the data results of the model are split into two mutually exclusive groups from the original input data set. In this research, we used this type of classification to distinguish assessment and non-assessment cases.

2.3.3 Supervised learning classification algorithms

2.3.3.1 Random Forest

a) Decision trees

The Random Forest algorithm's precursor is the decision tree algorithm. To increase accuracy and avoid "overfitting," decision trees first construct a decision tree, which is then followed by a pruning step that involves removing subtrees from the decision tree [52]. When creating a decision tree, the training data set is split recursively according to a locally optimal criterion until the majority of the records in each partition have the same class label [54]. Tree pruning is the process of removing leaves and branches that classify a single or very small number of data vectors from a decision tree in order to increase the tree's capacity for generalization [54]. When creating a tree, the splits for each element are first evaluated, the best split is chosen, and partitions are then created using the best split. Partitions can then be made utilizing by applying the splitting criteria to the data after the optimal split has been identified. There are several available dividing strategies. Some of the common schemes are entropy and the gini index. If a data set S contains examples from m classes, the Entropy(S) and the Gini(S) are defined as followings:

$$Entropy(S) = -\sum_{j=1}^{m} P_j \log P_j \qquad (1)$$

$$Gini(S) = 1 - \sum_{j=1}^{m} P_j^2 \qquad (2)$$

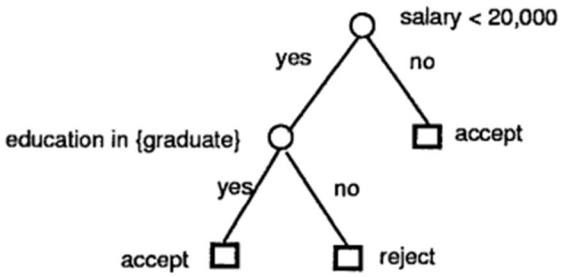
[54]

Where P_j is the relative frequency of class j in S. If attribute A is used to divide the data set S, we may calculate the information gain based on the entropy or the gini index [54]. The measure of disorder, uncertainty, and randomness in a closed atomic or molecule system is a common definition of entropy in molecular physics. For instance, if a system's entropy is large, it can be challenging to forecast the states of its atoms [55]. In other words, a high value of entropy means that the randomness in your system is high, meaning it is difficult to predict the state of

atoms or molecules in it. On the other hand, if the entropy is low, predicting that state is much easier [56]. The machine learning field operates on the same basis. In the field of machine learning, the disorder of the data processed inside the project is measured. In machine learning, the lower the entropy, the more straightforward it is to derive meaningful conclusions from a given set of data [56]. The Gini Index, sometimes referred to as Gini impurity, determines the likelihood that a certain feature would be erroneously classified when chosen at random [57]. The values of the Gini index range from 0 to 1, where 0 represents the purity of classification—that is, all the elements are members of a single class or there is only one class present. The random distribution of pieces among the different classes is shown by 1. The Gini Index's value of 0.5 indicates that the elements are distributed equally across some classifications [57]. When creating the decision tree, the features with the lowest Gini Index value would be given priority [57]. We obtain improved categorization at each node as the tree descends because the amount of impurity and uncertainty decreases. Rastogi and Shim [52] presented a simple example of how decision trees work with visualisation of the tree. The example demonstrated how the classification model used two attributes—salary and education—to decide whether or not to grant a loan to an applicant. The classifier's objective was to infer, from the training data, brief and significant wage and educational requirements that determine whether a loan request is approved or denied. The sample training data and diagram below helped to illustrate how the model would come up with a prediction.

salary	*education*	label
10,000	high-school	reject
40,000	under-graduate	accept
15,000	under-graduate	reject
75,000	graduate	accept
18,000	graduate	accept

(a)



(b)

Figure 4 Sample illustration of decision trees by Rastogi and Shim [51]

The decision tree in the example starts by determining the result of the first decision, which is a check if the salary is less than 20,000. If the condition is false, then the loan application is accepted without any further decisions. The assumption of this decision is that an individual with a salary of 20,000 and able is capable of repaying the loan if they were to default, it would be most likely easy to recover the payment, hence the individuals that fall into this categories are considered safe. If the salary is less than 20,000, the application is subjected to another

decision in order to determine their ability to pay back the loan. This decision is their level of education, which from the sample training data is either high-school or under-graduate. If their level of education is high school, the application is rejected. However if the level of education is undergraduate, then the application is accepted. The assumption in this case would be that despite their salary being under 20,000, they have a higher chance of paying back due to their current position and other factors such as the potential to have an increase in their income because they have attained a higher level of education. These decisions and assumptions are real-life decisions that a loan approving officer may make and the model attempts to replicate to make decisions, which therefore brings the human aspect into the automated computer decision-making process.

b) Random Forest

In order to lessen the workload associated with data gathering and increase efficiency, prediction modelling typically seeks to minimize the number of variables required to produce a prediction. [58]. A popular supervised machine-learning technique for categorization issues like the one we study is called random forest. An ensemble classifier called a random forest (RF) classifier creates several decision trees using a subset of training samples and variables that is chosen at random [59]. Random forests are particularly strong since they have several trees. The random forest technique builds decision trees using samples of data, obtains predictions from each one, and then uses voting to determine which is the best answer [60]. The method, which averages the predictions from many randomized decision trees, has demonstrated exceptional performance in scenarios when the number of variables is significantly greater than the number of observations [61]. The "divide and conquer" strategy, which is straightforward but effective, guides the operation of this algorithm. Sample portions of the data are used to generate randomized tree predictors on each little piece, which are then pasted (aggregated) together [61]. A subset of the original collection of attributes, chosen at random, is used by each decision tree node [62]. Among well-liked machine learning techniques, random forest offers a special blend of model interpretability and prediction accuracy. With the use of random sampling and ensemble techniques, RF is able to produce more accurate forecasts and improved generalizations [63]. As the number of trees grows, it does not always mean the performance of the forest is significantly better than previous forests (fewer trees), and doubling the number of trees is worthless. It's also feasible to say that, absent a massive processing environment, there exists a threshold above which no appreciable

improvement occurs [62]. The figure 13 below shows an illustration of the general working of the random forest algorithm.

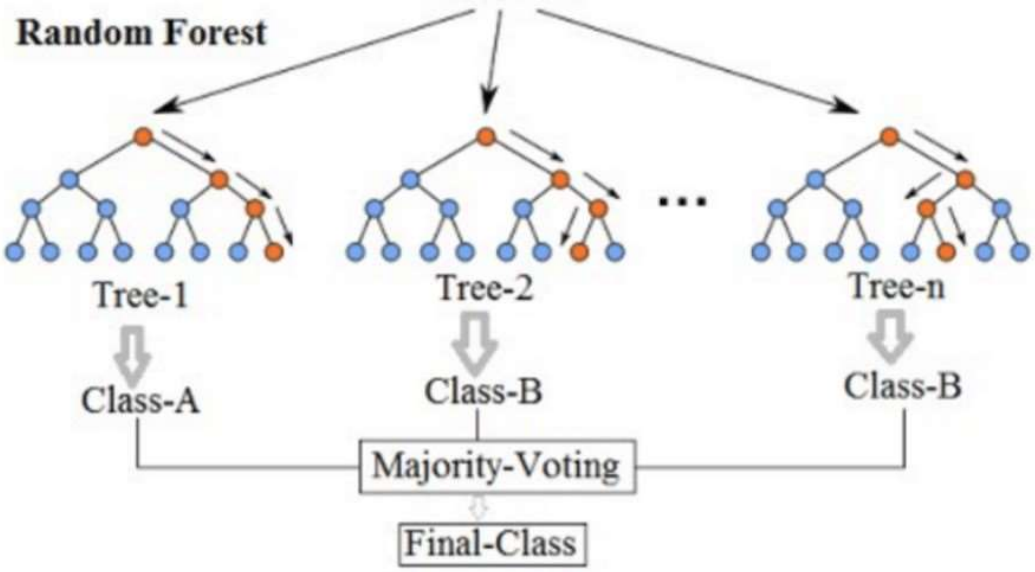


Figure 5: Random Forest Illustration [68]

A Random Tree is a tree drawn at random from a set of possible trees, with m random attributes at each node. The term "at random" means that each tree has an equal chance of being sampled. Random Trees can be efficiently generated, and the combination of large sets of Random Trees generally lead to accurate models [62].

The Strong Law of Large Numbers and the tree structure dictate that, for a large number of trees, very certainly all sequences converge to the following as the number of trees increases:

$$Px, y(P_{\Theta}(h(X,\Theta) = Y) - max\ P_{\Theta}(h(X,\Theta) = j) < 0) \quad (3)$$

[64]

The Gini Index, which gauges an attribute's impurity in relation to the classes, is used by the random forest classifier as an attribute selection metric [65]. For a given training set T, selecting one case at random and saying that it belongs to some class, the Gini index can be written as follows:

$$\sum \sum_{j \neq i} (f(C_{\mathrm{i}}, T)/|T|)\ (f(C_{\mathrm{j}}, T)/|T|) \quad (4)$$

[65]

Consequently, N trees make up the random forest classifier, where N is the number of trees that must be produced and can be any value that the user specifies. Every dataset case is handed down to every N tree in order to classify a new dataset. In that instance, the class with the greatest votes out of N is selected by the forest [65].

The process of the random forest algorithm can be broken down into five major stages. These are sample selection, decision tree construction, sample prediction, voting and prediction.

Stage one – Sample selection

In the first stage, random samples are obtained from the original training dataset N number of times, where N is the number of trees to be grown, which can be any value defined by the user.

Stage two – Decision tree construction

Using every random sample that was collected in the first step, a decision tree is created in this stage.

Stage three – Sample prediction

In this stage, each tree makes an independent prediction and give the results as output.

Stage four – Voting

When all the sample have made their prediction, the algorithm them conducts a voting process on all the results in order to find the best result.

Stage five – Prediction

In the fifth stage, the prediction with the highest votes is picked and used for the final prediction.

The diagram below illustrates the process by which a prediction is made using random forest from the training dataset to the random splitting to the voting and finally the prediction.

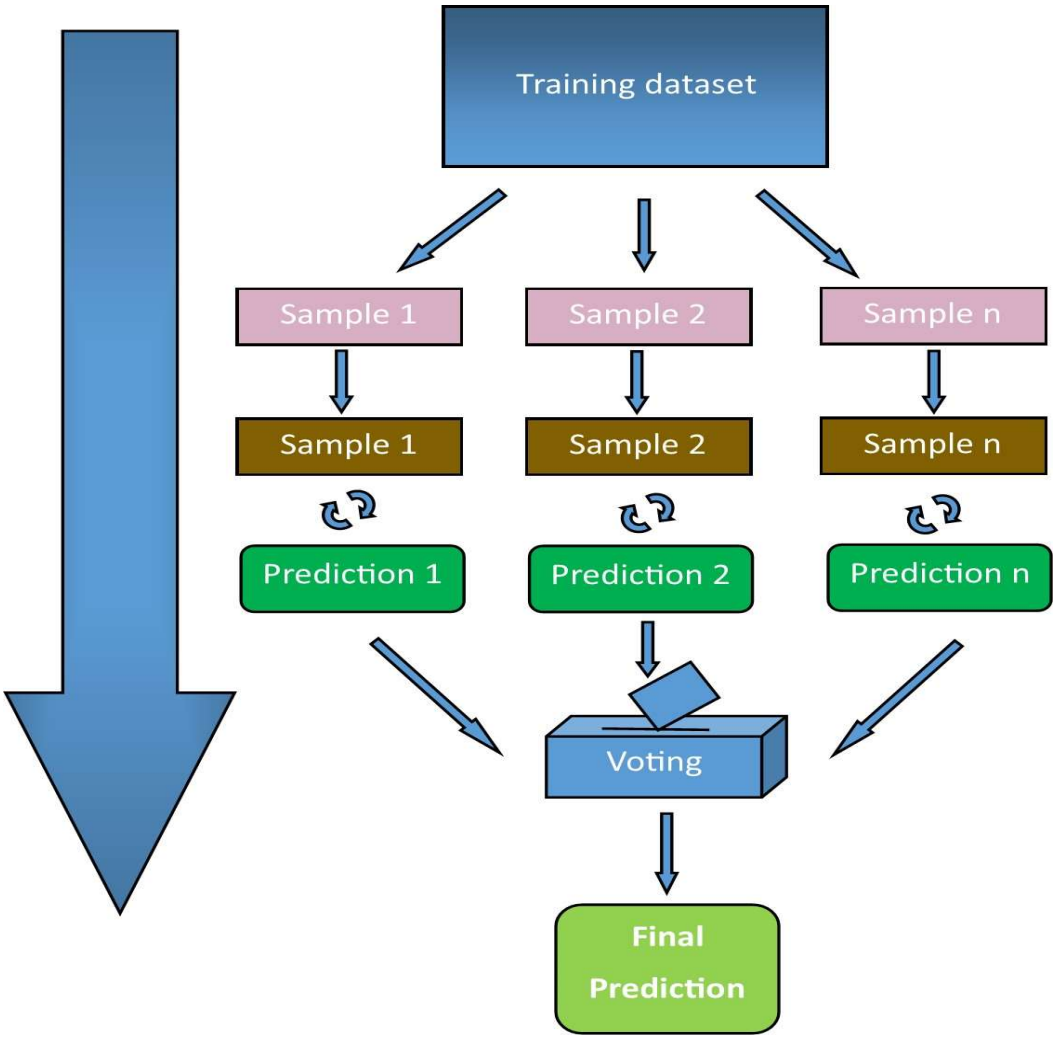


Figure 6: Random Forest stages

Before training is complete, three primary hyperparameters for Random Forest must be specified. These three factors are the number of features, the number of trees, and the size of the nodes. Other variables that could be taken into account when setting up and refining the RF model include max_depth: The maximum depth of any decision tree is determined by this option. While a lower max_depth can result in underfitting, a larger max_depth can cause overfitting. The value of the min_samples_split argument indicates the bare minimum of samples needed to split an internal node. An increased value may aid in avoiding overfitting. The parameter min_samples_leaf determines the bare minimum of samples that must be present at a leaf node. An increased value may aid in avoiding overfitting. max_features: The maximum number of features to take into account while determining the optimal split is indicated by this option. Less is more when it comes to overfitting. The number of decision trees that will be generated in the forest is indicated by the parameter n_estimators. Large

dataset classification issues are best suited for random forest, and the model's performance can be increased by adjusting a few of its parameters.

2.3.3.2 Support Vector Machine

The supervised learning method known as Support Vector Machine is primarily employed in classification tasks. Although it is particularly suitable for classification problems, it may also be used to regression difficulties. Many data scientists find support for SVM, an algorithm that is widely used and produces significant accuracy while requiring less processing resources than other algorithms. The objective of a support vector machine is to locate the hyperplane in an N-dimensional space that clearly classifies the data points. Out of all the potential hyperplanes, the classification looks for the optimal one to divide two classes. It achieves this by calculating the greatest separation between the data points in the two classes. It heavily depends on the data points that are in closer proximity to the support vectors hyperplane. The points aid in defining the SVM model because they have an impact on the hyperplane's orientation.

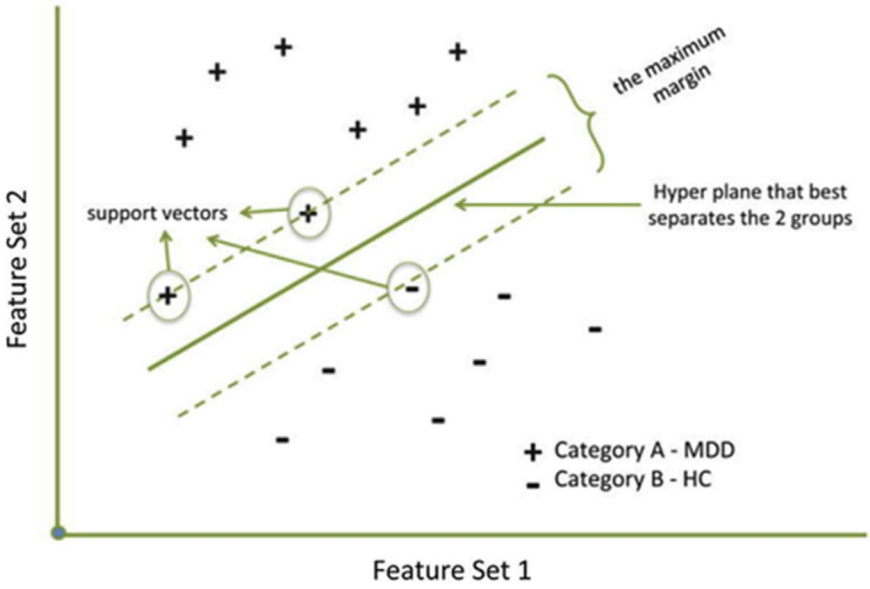


Figure 7: Illustration of SVM hyperplane [82]

The SVM chooses the maximum margin splitting hyperplane if we define the margin of the hyperplane as the distance between the separating hyperplane and the closest expression vector. By choosing this particular hyperplane, the SVM's capacity to predict the right categorization is maximized [66]. Ignoring outliers in favour of identifying the optimal hyperplane that maximizes margin is one of the SVM's properties. When compared to other classifiers, SVM's strength and reputation primarily stem from its capacity to achieve consistent, accurate, and

broadly applicable results—even in situations where the dimensionality of the feature space greatly surpasses the quantity of training sample observations [67]. The optimal performance of the SVM algorithm greatly hinges on its parameter settings. To create a suitable model, the support vector machine's two primary parameters—the soft margin parameter and the kernel function free parameter—need to be adjusted [68].

2.3.3.3 Adaptive Boost (AdaBoost)

Another popular supervised learning approach for classification tasks is called adaptive boosting, or AdaBoost. Freund and Schapire created AdaBoost, the first boosting algorithm used in practice, in 1995. It leverages on other weaker learners by combining them to make a stronger learner. Boosting algorithms operate on the premise of first building a model on the training dataset and then developing a second model to correct the errors in the previous model. This procedure enables AdaBoost to concentrate on the samples that were incorrectly categorized [69]. Until the mistakes are minimized and the dataset can be forecasted with accuracy, the process is repeated. The error rate and the number of iterations decide when the loop ends. When the training reaches the required number of iterations, the learning is often finished [69]. Below is a schematic that shows this procedure..

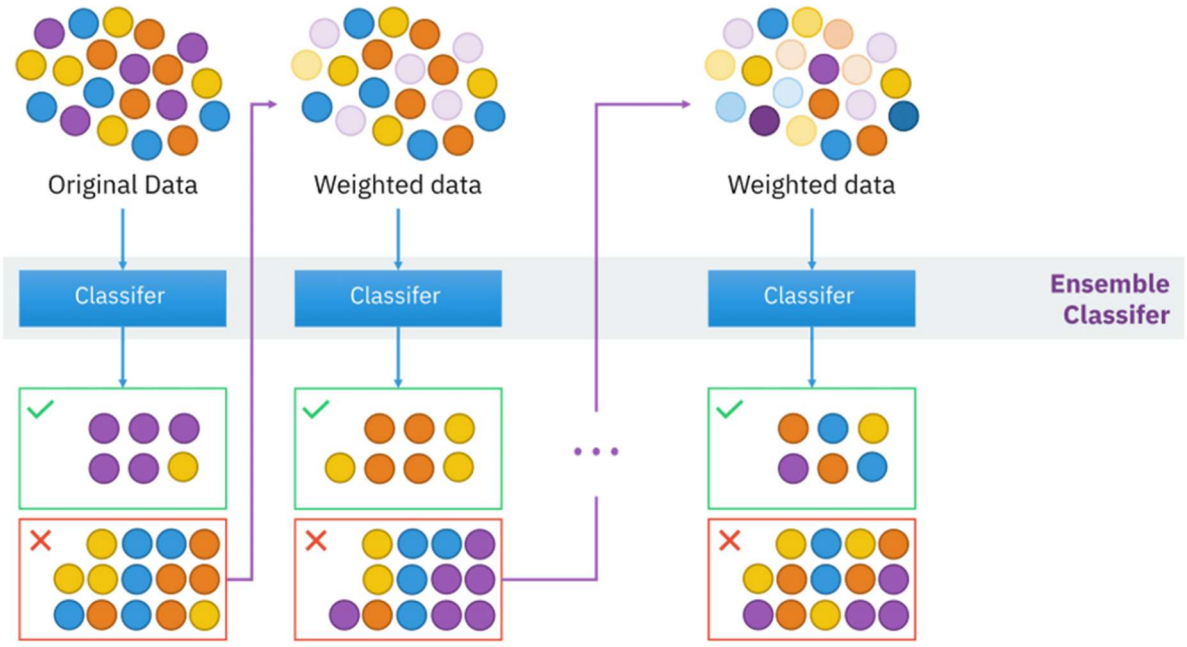


Figure 8: Boosting algorithm building process [85]

One of the reasons AdaBoost algorithm produces good results is the diversity and variety among weak classifiers [70]. Diversity can be defined as the notion of differences between the outputs of each classifier. However, neither a formal definition nor a diversity evaluation is

exact. Ludmila [71] discovered multiple diversity metrics. It is possible to adjust the model's development to increase accuracy and overall performance. For the model to function properly, a large number of trees—typically hundreds—must be included [72].

2.3.4 Machine learning in Taxation

Many governments find it extremely difficult to collect taxes for a variety of reasons. In many countries, visibility of taxpayers is very difficult to achieve. Therefore, the population that covers the undetected taxpayers tends to be small. This means that the small population that is visible tends to be stressed. Tax evasion is also a major problem with many tax collection authorities around the world. Other factors such as uneducated tax community, poor policies, cost of collection of taxes and tax authority workforce required to meet the demands of the collection contribute to the many challenges. In an effort to address the difficulties facing tax collection agencies, authorities have turned to data mining, machine learning, and artificial intelligence throughout the years. Authorities examined the ways in which these tools could support their data analysis and decision-making to enhance tax compliance, identify instances of tax evasion, and enhance internal operations and customer service in general.

Systems with artificial intelligence communicate with and adjust to their surroundings. These environments could include humans or intelligent systems, among other things [73]. Intelligent systems receive input from this environment and are able to behave intelligently according to the machine learning algorithms that were used to teach them during their learning phase. Human beings may change their reactions from their routine reactions and responses when presented with new information. This intelligent behavioural change is very critical to achieving success in dealing with taxpayer compliance. Accurately achieving acceptable levels of intelligent behaviour from computers is the aim of data mining, machine learning, and artificial intelligence taxation. Because taxes can be highly complicated, the systems that are designed must be sophisticated enough to understand how the input is changing and respond in a way that is appropriate. Input data could come from various sources including internal organisation data, trading companies, other government institutions. Although tax professionals use the term "analytics" often, it is a general phrase that covers a wide range of topics, including business intelligence, dashboards, predictive and prescriptive tax analytics, and more sophisticated fields like artificial intelligence, data mining, and machine learning [73].

2.4 Related Work

Henrik Hoglund [74] proposed a decision support tool for the prediction of tax payment defaulters using genetic algorithm-based variable selection. His dataset comprised limited liability companies in Finland that were in arrears on value added taxes or employer contribution taxes. He discovered that factors assessing trade payables' solvency, liquidity, and payment terms were crucial in forecasting tax defaults.

One of the holes in this paper's analysis was that, while the research focused on taxpayers who have defaulted, defaulters might actually come from anyone who has been paying taxes, making it impossible to make a precise prediction. Even while taxpayers are filing and paying taxes on a regular basis, their filings may be irregular or declining, which could indicate that their firm is under hardship or engaging in tax evasion. Additionally, taxpayers who file a zero return out of ignorance or to fulfil their self-declaration requirements will not be included. Our study filled up the gaps in the literature by expanding the training and testing data's coverage.

E. Kıral and C. Mavruk [75] researched on trying to develop a model to predict tax declarations using Markov chain model. By building probability matrices of the transitions between classes that are described for each model, declaration probabilities for the year 2017 were estimated. By measuring the total of each model's mean square errors, the optimal Markov model was identified. Research focused on the tax declaration rates as important factors.

The gaps identified in this paper was to get better results, the research could have considered expanding the parameters that could affect the predictions and also used multiple prediction algorithms and evaluate the best performing model. Our research included more parameters that are likely to affect the results that were not considered in this research. Our research also evaluated three algorithms to determine which algorithm performs the best.

A research was conducted to try to utilize machine learning techniques to forecast the tax audits' deterrent effects by M. Rabasco and P. Battiston [76]. Using tax declarations data, they were able to forecast a rise in income of the future declarations by taxpayer after they had undergone a targeted. They discovered that when it came to identifying taxpayers who are more likely to increase their declarations, flexible models like classification trees and ensemble approaches outperformed linear models like Lasso and ridge regression.

The gaps identified in this paper were that the research focused on tax revenue likely to be collected from the taxpayers. It did not look at taxpayer who were likely to default and

subsequently undergo audits and assessments. The research also focused on small population of audited taxpayer, leaving the potential from the larger unaudited population. Our research built up on this taking into account the important variables required such as the taxpayers who are likely to default.

Using the tax payers for businesses in Armenia who use a conventional tax system, V. Baghdasaryan, H. Davtyan, A. Sarikyan and Z. Navasardyan [77] used machine learning tools to create a fraud prediction model, with gradient boosting serving as the primary method. For their research, they collected and used historical fraud and audit data, the administrative cost share, and external economic activity. The model was developed using scant information on the factors influencing the possible tax base and a small sample of taxpayers (observations).

The gap identified in this research was that the inclusion of more detailed, transaction-level data, which would undoubtedly improve the analysis, is not included in the study. Our research took into account the transaction level declaration data to get better results of the predictions.

An investigation was conducted to show how data mining and machine learning could improve public programs' efficacy and provide information for policymakers in Italy. The study focused on the tax rebate program that was implemented in that nation in 2014. In the event that a taxpayer has paid too much tax during a business process—like manufacturing—they can obtain a tax rebate, which lowers their tax bill. Enhancing a nation's competitiveness in international markets and, in certain situations, preventing double taxation on export goods are the primary goals of tax rebate policies [78]. The purpose of the study was to demonstrate the influence of machine learning algorithms and how they could improve the efficacy of the scheme's beneficiary selection process. Since the recipients would be determined by a machine learning method that is simple to understand, this would aid in transparency [79]. The findings demonstrated that, as the monies were directed into households not specifically targeted by ML, 29.5% of the policy's allocated funds, or roughly 2 billion euros, could have been saved without lowering overall consumption expenditure [79].

The gap identified in this research was that the model was developed on a rebate benefit program which tends to have different variable and features compared to a tax assessment and audit identification system. Our model was developed as a build up to this research, leveraging on the concept for use on tax declaration data.

Hannah Seippel [80] analysed machine learning models to predict customer purchases in retail stores. The study's objective was to classify whether or not a German clothes retailer's online

shopping experience would result in a purchase. The study employed a variety of data types, including static and dynamic data, to determine the most accurate way to categorize users as either purchasing or not. This analysis was envisaged to help the company monitor their performance in order to improve their services to their customers. This information could also help the company with personalization of their customers. For example from the results, they improve the customer experience by offering them certain popular items based on previous successful sales. It would also give the company a competitive advantage over others. Several prediction models were used to the sequential clickstream and the static customer data in the study. It compares the model's performance in the end. The Cross Industry Standard Process for Data Mining (CRISP-DM) approach served as the foundation for the study. Boosted tree, Random Forest, Support Vector Machine (SVM), Feed-forward Neural Network (FNN), Logistic Regression (LR), and Recurrent Neural Networks (RNN) are some of the classifiers that were employed. The results showed that the Random Forest model performed best of all the classifiers.

The gap identified in this research was that the research was carried out limited to the retail industry and therefore the model produced could not be used in a tax assessment prediction problem in its form and a lot of tax related features are overlooked.

Li Ying [81] offered a useful experimental foundation for bank credit approval, which used data mining classification algorithms to separate risky consumers from a huge pool of loan applicants. Using the Python programming language and the Random Forest, Logistic Regression, and SVM classification techniques, he created bank credit default prediction models. Using the five model effect evaluation statistics—accuracy, recall, precision, F1-score, and ROC area—he compared the classification effects of the classifiers. The features of risk clients were identified by the classification models, which were built using the pertinent biographical data and consumption information of previous loan applications. His analysis of comparisons The findings of the experiment indicate that the Random Forest method performed better for the bank credit default precision model than the SVM and Logistic Regression classification algorithms. The dataset used was small hence may have not been very conclusive as required because quantitative data analysis requires a large dataset for better modelling and results.

The gap identified in this research was that the research looked at customer data from the banking industry analysing their transactional trends but did not take in to account customer

declaration trends and variables that could have an impact on the model's overall performance. Our research used some of the concepts of this research to build a prediction model suited for the tax administration industry.

Pompe and Bilderbeek [82] examined how individual financial measures and bankruptcy models could be used to forecast bankruptcy. They anticipated a declining trend, which would show up in the activity and profitability ratio values first, then the solvency ratio values, and finally the liquidity ratios prior to bankruptcy. They discovered that there was no set sequence for when the various financial ratio categories become predictive. Similar predictive efficacies were observed in ratios assessing several aspects of a firm's financial status five years prior to failure. The insolvency of start-up companies, which was harder to anticipate than that of well-established companies, was the subject of the other theory that was investigated. The outcome corroborated this theory by the result and it was concluded that separate models for younger firms and one for established firms would have to be used instead of a general model for all the firms.

Sahar Sabbeh [83] examined and evaluated the effectiveness of many machine-learning approaches that may be used to forecast which consumers were most likely to discontinue doing business with a company. He investigated various machine learning system models to analyse behavioural and personal data of consumers in order to provide businesses with a competitive edge through higher customer retention rates. The client database of a telecommunications firm served as the dataset for this study's tests. Customers' statistical data, including 17 explanatory factors about their service consumption during the day, international calls, and customer support calls, were included in the dataset. He employed eleven analytical strategies that fall under several learning domains. Discriminant analysis, decision trees (CART), support vector machines, logistic regression, instance-based learning (k-nearest neighbours), ensemble-based learning (Random Forest, Ada Boosting trees, and Stochastic Gradient Boosting), Naïve Bayesian, and multi-layer perceptrons are some of the approaches that have been selected. After evaluating the classifiers, it was discovered that ADA Boost and random forest produced the best accuracy results, both at 96%. The outcomes additionally shown that, with 94% accuracy, multi-layer perceptrons and support vector machines could also be advised.

The gap identified in the research was that the model generated could only be used in a closed industry and could not be extended to others. Our research used this concept do develop a model for the taxation industry.

Yilmazer and Kocaman [84] attempted to address the extremely complicated issue of development by employing novel methods of mass real estate appraisal that leverage artificial intelligence and machine learning to assist Turkish municipalities in precisely valuing tax properties in comparison to taxpayer declarations. In an urban residential area with available business properties, they investigated a mass appraisal option. To create the model, they employed random forest, another automated machine learning technique, together with linear multiple regression. 1162 instances were randomly divided into training (80%) and validation (20%) subsets for the random forest model [84]. The number of trees to be utilized in the model was determined by using the out of bag error (OOB), an indicator and important measuring tool. 100 trees were chosen. It was concluded that the random forest algorithm yielded marginally superior results to the linear multiple regression method for mass appraisal and was able to describe the dependent variable with accuracy [84]. Additionally, they came to the conclusion that the residential assessment approach's outcomes might be used to Turkey to rectify declared tax values, since at the time, municipalities determined property tax values by using taxpayers' declarations.

The gap identified in the research was that the model could only work for predicting taxpayer declaration values in terms of the tax amount. It did not look at identification of under declarations and the classification of these taxpayers. Our research builds on this by focusing on classification of assessment and non-assessment declarations.

A Lismont [85] study used social network analytics to forecast tax evasion. They created a network of companies linked by common board membership in order to create a prediction model. They then used decision trees, random forests, and logistic regression as three analytical tools to develop five models that used either firm characteristics, network characteristics, or various combinations of both. The model with the strongest prediction capacity for tax evasion was determined to be the random forest, which includes firm characteristics, network characteristics of firms, and network characteristics of board members [85]. They came to the conclusion that financial experts and authorities may utilize their knowledge to forecast which companies are most likely to have low taxes and to be in danger.

The gap identified in the research was that the research only focused of tax avoidance and overlooked other significant categories such as under declarations, tax evasion and fraud. Our research builds up on this and include other relevant features.

In order to evaluate the random forest classifier's performance with that of support vector machines (SVMs) in terms of classification accuracy, training time, and user-defined parameters, Pal [65] conducted a study. For the purpose of training and testing the model, they employed Landsat Enhanced Thematic Mapper Plus data from a region of the United Kingdom with seven distinct land coverings. According to the study's findings, the random forest classifier outperformed SVMs in terms of training time and classification accuracy. This study also found that random forest classifiers require fewer user-defined parameters than support vector machines (SVMs), and that these parameters are also simpler to construct [65]. The gaps identified in the research was that the research looked to compare general model performance in efforts to identify generally model that perform best. This research helped us identify the possible algorithms to use to build our model.

2.5 Chapter summary

Random forest is a potent machine-learning algorithm that has been more well-known recently because of its exceptional performance in a variety of tasks, including feature selection, regression, and classification. It is an ensemble approach that creates a more reliable and accurate model by combining several decision trees. By using this method, the chance of overfitting is decreased and a more stable model that fits new data well is produced. Random forest has many over other models for prediction of classification problems due to its superior performance, adaptability to different types of data, interpretability, and scalability. Its ability to handle messy data, produce accurate predictions, and provide insight into the underlying data make it an excellent option for a variety of machine learning jobs. Other model algorithms such as the support vector machine and Ada boost are good performing models for classification problems because of the way they are designed. From the literature review we were able to select the 3 classifiers for use and evaluation as they have yielded positive results in past similar research projects. In the research we developed prediction models based on the three and compared and evaluated their performance in order to find the best performing algorithm.

3. METHODOLOGY

3.1 Introduction

In the previous chapter we looked at taxation in Zambia, the various tax type administered how audits are conducted and how the final assessment is arrived at and posted on the taxpayer's account. We looked at how machine learning and artificial intelligence can be used to enhance business processes and provide a more intelligent and automated processes that add value to organisations. We finally looked at and evaluated at other research in taxation and other similar industries where machine learning and artificial intelligence was used to create solutions to similarly structured problems. From the evaluations we did in the literature review section, we were able to get some insight into the possible machine learning approach that would be suited for our kind of problem among classification algorithms. In this section, we looked at the identification of the data required to build our model. We discuss in detail the methodology used in the data mining and implementation of the research project. We describe the procedures we followed in gathering, processing, and using data to build a model for identifying tax declarations for tax assessments. The Cross Industry Standard Process for Data Mining (CRISP-DM) approach was applied.

3.2 CRISP-DM Methodology

The industry-standard and cross-industry process model for implementing data mining initiatives is called CRISP-DM [86]. The approach was initially developed in 1996 as a common technique for initiatives including data mining, analytics, and data science. The technique outlines a loosely structured six-phase sequence that enables the development and application of a data mining model in an actual setting [87]. An outline of a data-mining project's life cycle can be found in the CRISP-DM reference model. It includes the stages of a project, their corresponding assignments, and their final products [88]. For evidence mining, CRISP-DM is a useful tool because of several qualities. It provides a broad process model that captures the scope and general structure of the methodology [89]. The project cycle is divided into six stages: deployment, assessment, modeling, data preparation, business understanding, and data understanding. The steps in this process are not in a strict order, and it is typically required to go back and forth between them. At times a phase can move from one of the final stages such as evaluation to the first phase business understanding depending on the requirement. The diagram below illustrates these stages.

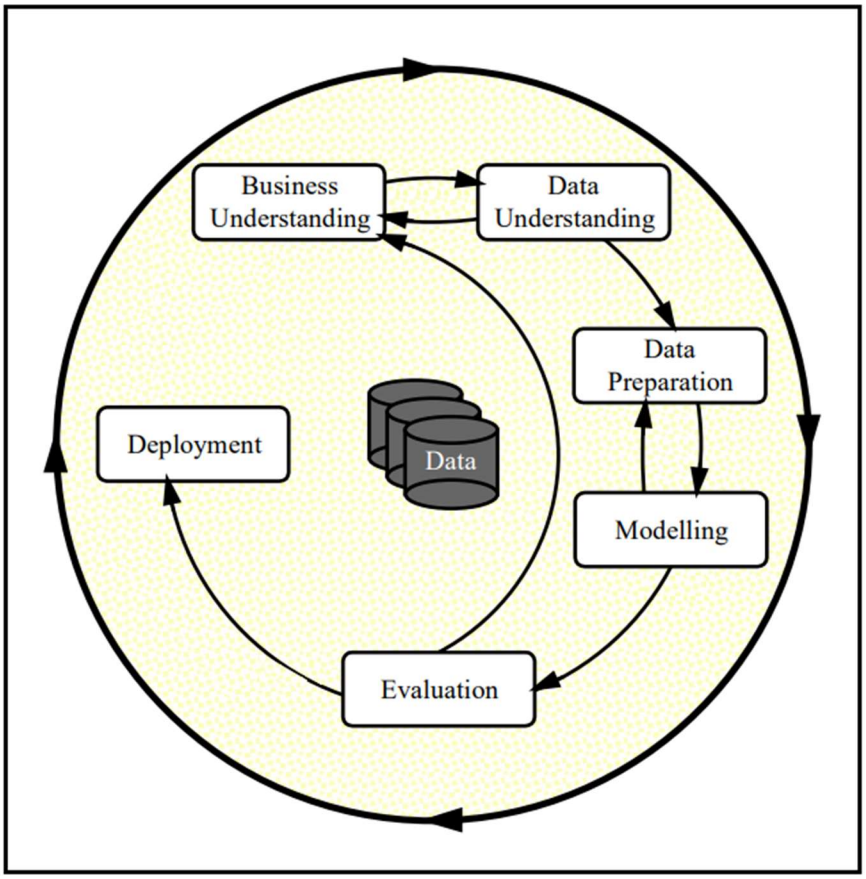


Figure 9: CRPSI DM process [66]

Development of the data mining and machine-learning project was done using the CRISP-DM methodology. The following are the activities carried out in each of the stages of the cycle from business understanding stage to the deployment of the model.

3.2.1 Business understanding

The business understanding stage is crucial in laying the foundation for a successful data-mining project, as it helps to ensure that the project is aligned with the business objectives, stakeholder needs, and available data sources. At this stage, we engaged the business user members who mainly consisted of tax auditors to get an understanding of the current process. This process was conducted through various interviews with the audit personnel who have vast experience with the challenges in tax collection. During the interviews, we were able to learn the challenges affecting tax compliance from the different types of taxpayers. We were also able to understand the process and identify the gaps in the existing process, which formed the basis of the research aims and objectives. The following section goes into further information about the current business process. We were able to ascertain from the engagements that the authority's goals were to decrease tax evasion, increase the efficiency of the tax assessment

process, and increase the accuracy of tax assessments. We obtained high level information on the areas affecting assessments based on their observations on previous audit and assessment activities and based on their experiences. They provided some of their observations in the return filing processes, payment and assessment processes. We used this high-level information to guide us on where we would obtain detailed technical system information from where we could collect information that the prediction model will utilize. We determined our source of the data as the tax administration relational database management system RDBMS. Finally, the business users determined the acceptable success factors in terms of how accurate our model was expected to achieve. The acceptable accuracy of a prediction model is subjective depending on the problem, the industry and many other implications. According to Barkved [90] , it is generally acceptable to have a prediction accuracy of about above 70%. We planned to develop a solution that was meant to enhance the existing process utilizing machine learning and data mining.

3.2.2 Data understanding

The data understanding stage in CRISP-DM is an important phase that includes the preliminary investigation and evaluation of the data to determine its quality, suitability, and relevance to the project's objectives. We collected all available relevant data that could have be relevant to the research. This data included the historical tax declaration data from the data source. With the knowledge we gained throughout the business understanding phase, we were able to extract the dataset needed for our model. As we further analysed the data and we were able to refine the data required through multiple iterations with business users in the business understanding stage and the data understanding stage. For training and testing purposes, we gathered sample data from the tax administration system database, which contained 199,999 records in 2020. The chart below shows Fig.10 the sample data comparison.

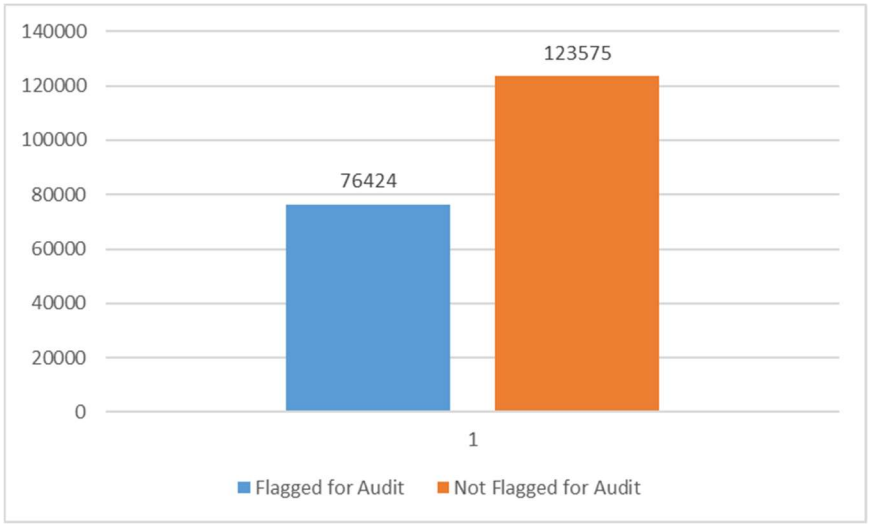


Figure 10: Chart showing comparison of sample data

Further to identify the connections and trends in the data, exploratory data analysis was done. These findings helped us, identify potential variables that could be used for training the model. Taxpayer declaration data that was divided into nine categories and included information about differences between input and output declarations was extracted. The features that were utilized to create the prediction model consisted of nine categories. The first of them is referred to as invoice reduction; these are transactions in which the taxpayer's declaration for a given invoice shows a decrease in the seller's sales invoices when compared to the purchaser's declaration for the same invoice. The second group is known as the "nil value on invoice category." This refers to transactions in which it was found that, although the buyer's invoice had a positive value, the seller's invoice had been declared as zero. The third group, known as the "nil declaration category," includes situations in which transactions were reported from the declarations of other buyers but the declaration as a whole was stated to be zero. The fourth category, known as the "no declaration category," refers to situations in which the buyer reported having made purchases from the seller during a specific time frame and the seller failed to file a declaration. The taxpayer's compliance with paying their taxes during the previous six months is indicated in the fifth category, which is six months payment compliance. The sixth category, one-year payment compliance, indicates the taxpayer's level of compliance with regard to filing returns and paying taxes during the previous year. The taxpayer's declaration compliance for the last six months is displayed in the seventh category, which is called six months declaration compliance. The eighth category, one-year declaration compliance, demonstrates the taxpayer's level of compliance with filing their tax returns during the previous year. The last category,

assessments in the last six months, indicates whether or not taxpayers were assessed during that time frame. The full details for the data used in the model development were discussed in the system implementation section of the document. The distribution of features is shown in Table 8 and fig. 11 below.

Table 8: Table displaying the distribution of features

Feature	Positive total	Negative total	Total flagged for audit
Invoice-Reduction	28898	171101	2724
Nil Value	20435	179564	14680
Nil Declaration	12671	187328	6826
No Declaration	122419	77580	72998
Payment compliance (6months)	99621	100378	37838
Payment Compliance (1year)	92662	107337	50195
Declaration Compliance (6months)	100001	99998	38039
Declaration Compliance (1 year)	95000	104999	46398
Assessments(6 months)	88916	111083	55134

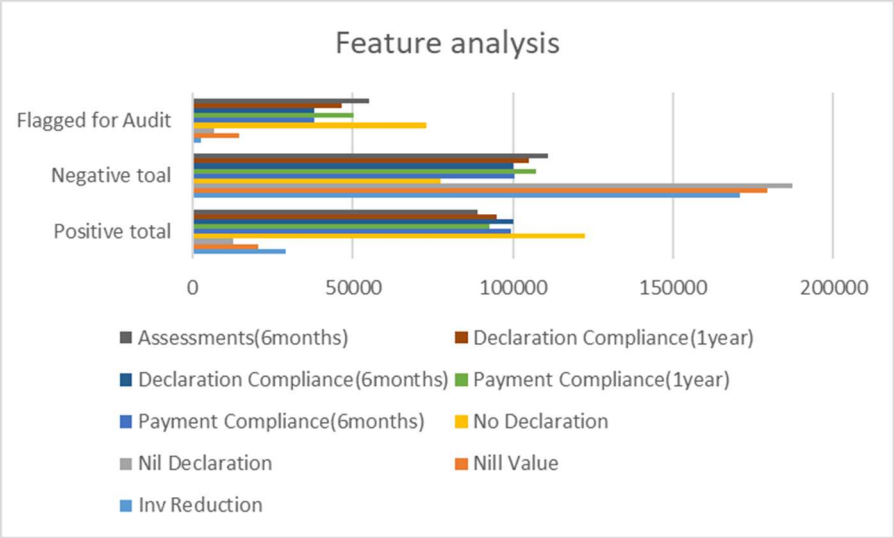


Figure 11: Diagram displaying the distribution of features

3.2.3 Data preparation

The stage of data preparation entails getting the data ready and ensuring that the data is in a format that can be easily analysed and used by the machine-learning algorithm to use to learn and make accurate predictions. The data was compiled and consolidated from the various and relevant areas of the data source into a single data set. In order to undertake data cleansing, we made sure our data had no null values. Data transformation was applied to the data, whereby new variables that were necessary were formed and poorly formatted data was changed to a format that the machine-learning system could understand. The document's system implementation section goes into greater detail on the data preparation procedure.

3.2.4 Modelling

The modelling technique we used was selected after evaluation and review of three algorithms: Ada Boost, Random Forest, and Support Vector Machine. In order to create a prediction model that can effectively and precisely evaluate tax declarations, modelling is a crucial step. To feed the machine learning algorithms that create models, the data was divided into training and test sets. The three models were developed and the parameters tuned and adjusted to get the most optimal results. The modelling process is discussed further in detail under the system implementation section of the document.

3.2.5 Evaluation

This assessment phase is utilized to determine how successful the predictive model developed and identifying areas for improvement. It involves evaluating the performance of the models on new data, assessing its impact on the tax assessment process, monitoring its performance over time and making the necessary adjustments. We evaluated the models performance after the models were developed using the score method of the ensemble library. We further evaluated the results using the confusion matrix, which gave us multiple dimensions of how the model performed. We also used the Receiver Operating Characteristic (ROC) curve, this shows us how the genuine positive rate (sensitivity) and the false positive rate (specificity) relate to one another. Finally we used the Logarithmic loss (LogLoss) to measure the model's performance. The full detail of the results are discussed in details in the results section of the document.

3.2.6 Deployment

The deployment stage involves preparing the model for deployment, integrating it into the existing tax assessment process, testing and monitoring the deployment. We developed a prototype web application using python language that consumed the prediction model we had created previously. Our application was hosted on the flask framework, which works as an application server for the application. The application we developed consisted of two integrated sub modules. The first module was a direct user application that has a user interface for entering transaction data by users and the prediction data response is displayed to the user. The second version was an API web service meant for direct integration which other external systems using standard API technologies. Once integrated, the external systems would be able to pass multiple requests to the API receive prediction results from the web service and use the data

for further processing in their respective internal system processes. More information on implementation is covered in the document's next section.

3.3 System Requirements specifications

3.3.1 Functional Requirements

This section highlights the system application functional requirements. These will define the specific functions and abilities that a system must hold to meet the desires of the users and system owners. The engagements we had with the business audit to find the challenges affecting compliance helped us to come up with the systems requirements that the prediction model was meant to address. The features of the model were derived after the evaluation of the challenges and review of the declaration data on the system. We outline functional requirements for the prototype application that uses the prediction model to make predictions on taxpayer's tax declarations and flag them for audit and assessments. The requirements cover the two modules of the system which include the user interaction module and the api module. The following are the functional requirements of the system.

Table 9: Functional Requirements

No	Functional Requirement	Mandatory / Optional
1.	The system should provide secure user authentication and privileges using a username and password to ensure that only authorized officers have access to the system.	Mandatory
2.	The system should only allow officers who have specific privileges to access specific sections of the system	Mandatory
3.	The system must be interfaced with the main tax administration system and allow users to use the same credentials that registered on the main tax administration system	Mandatory
4.	The system must be accessible from any web browser under the specified URL	Mandatory
5.	Sensitive information such as the user password and taxpayer personal information must be encrypted as they are transmitted over the network to the back-end servers	Mandatory

6.	The system should allow a user to run a prediction for a specific taxpayer using the TPIN and filter by the period from and period to	Mandatory
7.	The TPIN input boxes must have input validations	Mandatory
8.	The result of the prediction must be displayed for the user on the screen showing all the feature values for period selected	Mandatory
9.	The system must allow for prediction of multiple taxpayers for a selected period using a date picker	Mandatory
10.	The system should able to securely store transaction information while maintaining confidentiality and integrity	Mandatory
11.	The system must allow for real-time predictions on the data	Mandatory
12.	The user interface must only be available on the organisational intranet	Mandatory
13.	The API must be available over the internet	Mandatory

3.3.2 Non-functional Requirements

This section outlines the system application non-functional requirements. These will define express the qualities and features that the system must exhibit in the form of security, reliability performance, scalability and usability. We outline non-functional requirements for the prototype application that uses the prediction model to make predictions on taxpayer's tax declarations and flag them for audit and assessments. The requirements cover the two modules of the system which include the user interaction module and the API module. The following are the non-functional requirements of the system.

Table 10: Non-Functional Requirements

No	Non-Functional Requirement	Mandatory / Optional
1.	The system must comply with all relevant national and international data privacy laws	Mandatory
2.	Users must not be able to alter any information from the prediction results	Mandatory
3.	The system should have high availability	Mandatory

4.	They system should be able to interface with third parties using standard international technologies	Mandatory
5.	The system must be able to support multiple concurrent users	Mandatory
6.	They system must be integrated with the main tax administration system	Mandatory
7.	The system must be accessible from multiple locations seamlessly with centralized processing and a centralized database	Mandatory
8.	The system must respond to user input and interactions within an acceptable time	Mandatory
9.	The system must be able to recover from system failure without loss to data integrity	Mandatory
10.	The system must have an intuitive and easy to use user interface	Mandatory
11.	The system should be able to scale in accommodate increasing user and transaction activity	Mandatory
12.	The system must encrypt the payload information when sending to third parties through the interface	Mandatory

3.4 System Design and Implementation

3.4.1 Current business process

A tax assessment is a process of determining the tax amount to be paid by taxpayers where it is determined that a taxpayer failed to declare their self-assessment to the tax authority. The failure to declare takes different form but generally takes three forms. The first is failure to submit their declaration showing the business transactions for the period for their registered tax type as per period required. The second form is where the taxpayer actually submits their return but under declares their business activities and tax liabilities in an effort to lower their total tax obligations. The third form is where taxpayer submits a nil return, which means they had no activities in the period but they are simply meeting their tax obligations to avoid penalties and other charges. There are various reasons a taxpayer may fail to declare the correct taxes in their assessment or fail to declare all together. The most common reasons are tax

evasion and lack tax knowledge. A tax assessment is therefore an instrument used and enforced on taxpayers that have be found to have failed to declare the correct taxes.

In the current tax assessments business process, an audit process is commenced on the taxpayer were the taxpayer is suspected to have under-declared in a particular tax declaration. The tax authority may regularly undertake audits as a means of verifying compliance with a subset of taxpayers at any time. However, neglecting to fulfil tax responsibilities may result in a tax audit. [25].

a) There is a process in place for choosing audit cases. ZRA established selection criteria for audit cases, and that is what is adhered to [25]. Testing the veracity of the claimed or assessed tax is the main goal of tax audits [25]. When an audit process has begun, the taxpayer is communicated to and sent a notice of audit.
b) Before starting an audit, the authority will often ask the taxpayer for an appointment. The period, tax type(s), audit type, and books and records that are anticipated to be examined are all specified in the tax audit appointment letter. Thus, both before and during the audit, the taxpayer is required to provide all necessary books and records [25]. The auditor will outline the kinds of documentation that the taxpayer must submit as well as the intended audit methodology and procedures.
c) The authority may make an estimated assessment or use the information at its disposal to determine the tax owing and any interest owed on any income that has been omitted. Furthermore, failing to abide by particular tax laws or rules as prescribed by the various tax Acts may result in penalties [25].
d) The taxpayer is informed of the audit's conclusions following the audit. Before concluding the audit, the auditor notifies the taxpayer or their agent of any adjustments. [25].
e) After the audit is completed and the correct taxes have been calculated, an assessment is posted on the taxpayer’s account and an assessment notice is sent to them notifying them of the liability due for them to settle. It is worth noting that at the end of the audit, the assessment does not always increase their tax liability. In cases where it is found that the taxpayer may have over declared, the appropriate assessment will be posted to reduce their liability and a possible refund to the taxpayer.

The Diagram below Fig. 12 illustrates the different stages of the current audit process.

Figure 12: Current Audit process

All cases that require assessment go through the audit process and the assessment is the end product of the process. The challenges faced with the current approach is the large number of audit cases that auditors need to work on and some of which may yield little in terms of collections.

3.4.2 Proposed solution

The suggested methodology is intended to enhance the existing process of the final assessment. It is meant to reduce the number of audit cases manually selected and handled by auditors and reduce the level of effort in the audit process. Finally, it is meant to reduce the amount of resources required to conduct an audit and the consequent audit. This efficiency in resources and audit and assessment process will help the Zambia Revenue Authority become more efficient, effective and reduce on the cost of assessments. The proposed solution intends to find cases that yield the most or lead to the most revenue collection using data mining and artificial intelligence. The system will therefore suggest cases for audit that are likely to yield high revenue and cases that may automatically be posted directly to the taxpayer's account without human intervention the full audit process.

The system developed is made up of two main parts, both of which are containing process prediction logic and work together to produce the final integrated system. The first part is the binary classification prediction model that is developed using supervised learning and data collected from the existing system. The second part is a web-based application that uses the

logic from the prediction model, the existing system, therefore allowing the users to make configuration changes to the input data.

In the ZRA VAT tax system, sellers of products and services are expected to declare their invoices in their periodic declaration which will therefore be their output and the output tax will be calculated from there. Purchasers are expected to declare their purchases in their return as input therefore for every VAT transaction a purchase from one taxpayer is expected to have a corresponding sell from another taxpayer of the same item and value. Our prediction model adds further intelligence on which cases are likely to give highest yield assessment in order to maximise on the revenue collection. The proposed system attempts to avoid cases where auditors spend large amounts of resources on audit cases with less yield inequivalent to the cost of the audit process.

Using taxpayer declaration data that included variations between input and output declarations and nine categories, a machine-learning model was created. The first type of these transactions is known as invoice reduction; these are those in which the taxpayer's declaration shows a decrease in a seller's sales invoices relative to the purchaser's declaration for the same invoice. "Nil value on invoice category" is the name of the second category: In these transactions, it was found that the buyer's invoice had a positive value but the seller's invoice had been stated as zero. The third group, known as the "nil declaration category," includes situations in which transactions were reported from the declarations of other buyers but the declaration as a whole was stated to be zero. The fourth category, known as the "no declaration category," refers to situations in which the buyer reported having made purchases from the seller during a specific time frame and the seller failed to file a declaration. The taxpayer's compliance with paying their taxes during the previous six months is indicated in the fifth category, which is six months payment compliance. The sixth category, one-year payment compliance, indicates the taxpayer's level of compliance with regard to filing returns and paying taxes during the previous year. The taxpayer's declaration compliance for the last six months is displayed in the seventh category, which is called six months declaration compliance. The eighth category, one-year declaration compliance, demonstrates the taxpayer's level of compliance with filing their tax returns during the previous year. The last category, assessments in the last six months, indicates whether or not taxpayers were assessed during that time frame. The full details for the data used in the model development were discussed in the system implementation section of the document. The diagram below Fig. 13 illustrates the proposed solution.

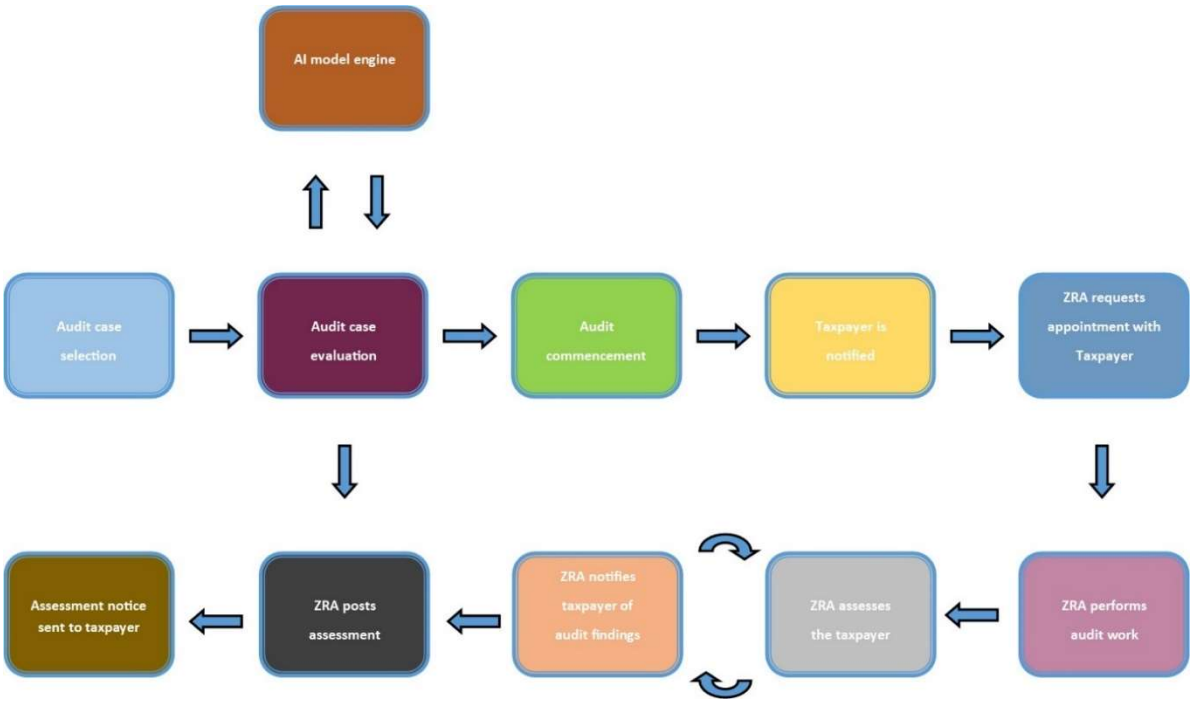


Figure 13: Proposed Audit process

3.4.3 Design specifications

This section offers a thorough explanation of the technical specifications used during the research. In the literature review chapter, we discussed and reviewed a number of machine learning algorithms and their applications. We also reviewed previous development works related to the problem we address in the document. After extensive review and analysis, we narrowed down algorithms to we used for our problem to three algorithms. To determine which algorithm would work best for the given situation, we created and evaluated three models based on the three algorithms, namely Random Forest, Support Vector Machine and AdaBoost. In order to accomplish our goal, we identified and utilized a data source and collected the required data from the various areas in the database. We also used different processing methods, as well as a range of modelling evaluation metrics. We provide more thorough explanations of the data source in this subsection, pre-processing techniques, feature engineering methods, modelling techniques, and evaluation metrics used in this project.

3.4.3.1 System architecture

Spacey [91] describes system architecture as the structural design of a system showing various layers, components and services working together as a system. The tax assessment data mining system application was designed as a web-based application with a client-server architecture.

The system comprised three primary elements: the front-end graphical user interface, the data processing engine, and the back-end database. In this section, we discuss the system architectural design and highlight how the entire system is set up.

3.4.3.1.1 Network design

The system architecture was designed for scalability and capacity for managing large volumes of information. The system was designed to work on a cloud-based platform, which provides high availability, fault tolerance, and good scaling capabilities. It was designed to be installed behind the internet firewall and configured a virtual private network for external connections. Internal staff should have access to the system via the internal internet. All connections to the application should be encrypted to ensure data integrity is maintained over the public internet. The diagram below Fig. 14 shows the network architecture.

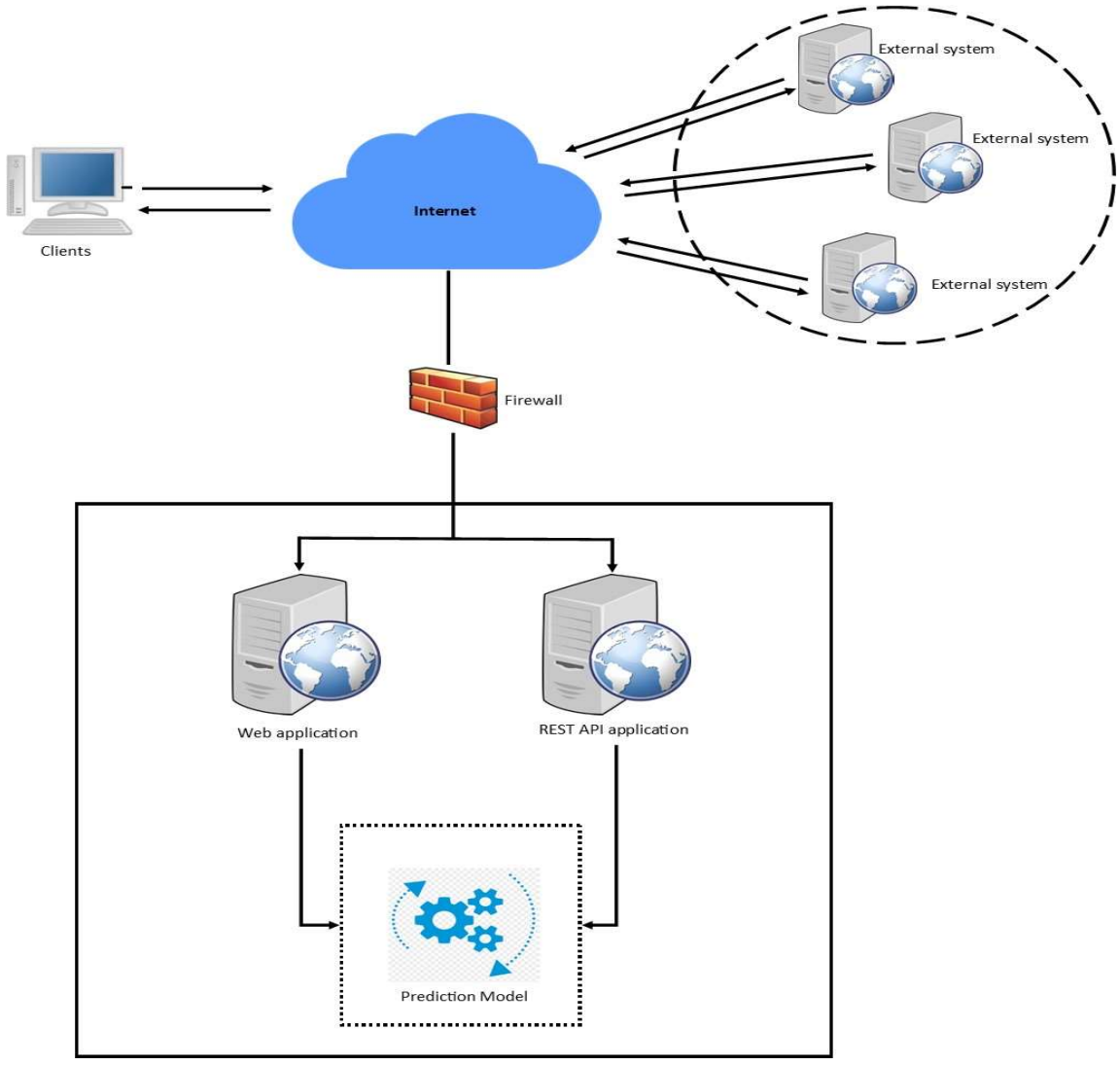


Figure 14: Proposed System architecture

3.4.3.1.2 Front-end design

The front-end interface, which was created with HTML, CSS, and JavaScript, enables graphical user interface (GUI) interaction between users and the system. The GUI is intuitive and user-friendly, with input fields for data entry and interactive data visualizations to display results. The system was designed using boostrap to have a responsive screen that responds adequately to differences in screen size such as tablets, phones and laptops in order to improve user experience.

3.4.3.1.3 Database Design

The back-end database was built using MySQL relational database which stored all the tax assessment and tax declaration data. The database was designed and normalised to the 3rd normal form to enable efficient data retrieval and analysis. We made use of database procedures and database jobs to help with automatic scheduling of tasks such pulling new declarations from the primary system for tax administration and preparing the data for the prediction model.

The following were the tables that were created for system.

Table 11: Table showing the user table

Table name	Column	Data type
user	id (PK)	int
	user_name	varchar(15)
	password	varchar(300)
	status	varchar(10)
	user_type (FK)	int
	created_date	datatime

Table 12: Table showing the user type table

Table name	Column	Data type
user_type	id (PK)	int
	type_name	varchar(15)
	type_description	varchar(300)
	type_status	varchar(10)

Table 13: Table showing the taxpayer table

Table name	Column	Data type
taxpayer	id (PK)	int
	user_id (FK)	int
	firstname	varchar(20)
	surname	varchar(20)

	company_name	varchar(50)
	tpin_no	int

Table 14: Table showing the officer user table

Table name	Column	Data type
officer_user	id (PK)	int
	user_id (FK)	int
	firstname	varchar(20)
	surname	varchar(20)
	user_type_id (FK)	int

Table 15: Table showing the admin user table

Table name	Column	Data type
admin_user	id (PK)	int
	user_id (FK)	int
	firstname	varchar(20)
	surname	varchar(20)
	user_type_id (FK)	int

Table 16: showing the gen_status table

Table name	Column	Data type
gen_status	id (PK)	int
	status_name	varchar(10)
	status_description	varchar(20)

Table 17: showing the user type table

Table name	Column	Data type
user_type	id (PK)	int
	type_name	varchar(10)
	type_description	varchar(20)
	type_status_id(FK)	int

Table 18: showing the privilege table

Table name	Column	Data type
privilege	id (PK)	int
	privilege_name	varchar(10)
	privilege_description	varchar(20)
	privilege_status_id(FK)	int

Table 19: showing the role table

Table name	Column	Data type
role	id (PK)	int
	role_name	varchar(10)
	role_description	varchar(20)
	role_status_id(FK)	int

Table 20: showing the privilege role table

Table name	Column	Data type
privilege_role	id (PK)	int
	privilege_id(FK)	int
	role_id(FK)	int
	privilege_role_status_id(FK)	int

Table 21: showing the user role table

Table name	Column	Data type
user_role	id (PK)	int
	user_id(FK)	int
	role_id(FK)	int
	user_role_status_id(FK)	int
	created_date	datatime

Table 22: Table showing the vat returns table

Table name	Column	Data type
vat_returns	id (PK)	int
	start_date	datatime
	end_date	datatime
	created_date	datatime
	taxpayer_id (FK)	varchar(20)

Table 23: Table showing the sales invoice table

Table name	Column	Data type
sales_invoice	id (PK)	int
	vat_return_id (FK)	int
	invoice_no	varchar(20)
	invoice_description	varchar(100)
	invoice_amount	double
	buyer_taxpayer_tpin	int

Table 24: Table showing the purchases invoice table

Table name	Column	Data type
purchases_invoice	id (PK)	int
	vat_return_id (FK)	int
	invoice_no	varchar(20)
	invoice_description	varchar(100)
	invoice_amount	double
	seller_taxpayer_tpin	int

Table 25: Table showing the nil returns table

Table name	Column	Data type
nil_returns	id (PK)	int
	vat_return_id (FK)	int

Table 26: Table showing the assessment data table

Table name	Column	Data type
assessment_data	id (PK)	int
	vat_return_id (FK)	int
	invoice_reduction	int
	nil_value_on_invoice	int
	nil_return	int
	no_return	int
	amount	double
	tpin	int
	created_date	datatime

Table 27: Table showing the prediction results table

Table name	Column	Data type
prediction_results	id (PK)	int
	assessment_data (FK)	int
	prediction	int
	prediction_date	datatime

The diagram below shows the entity relationship diagram for the database.

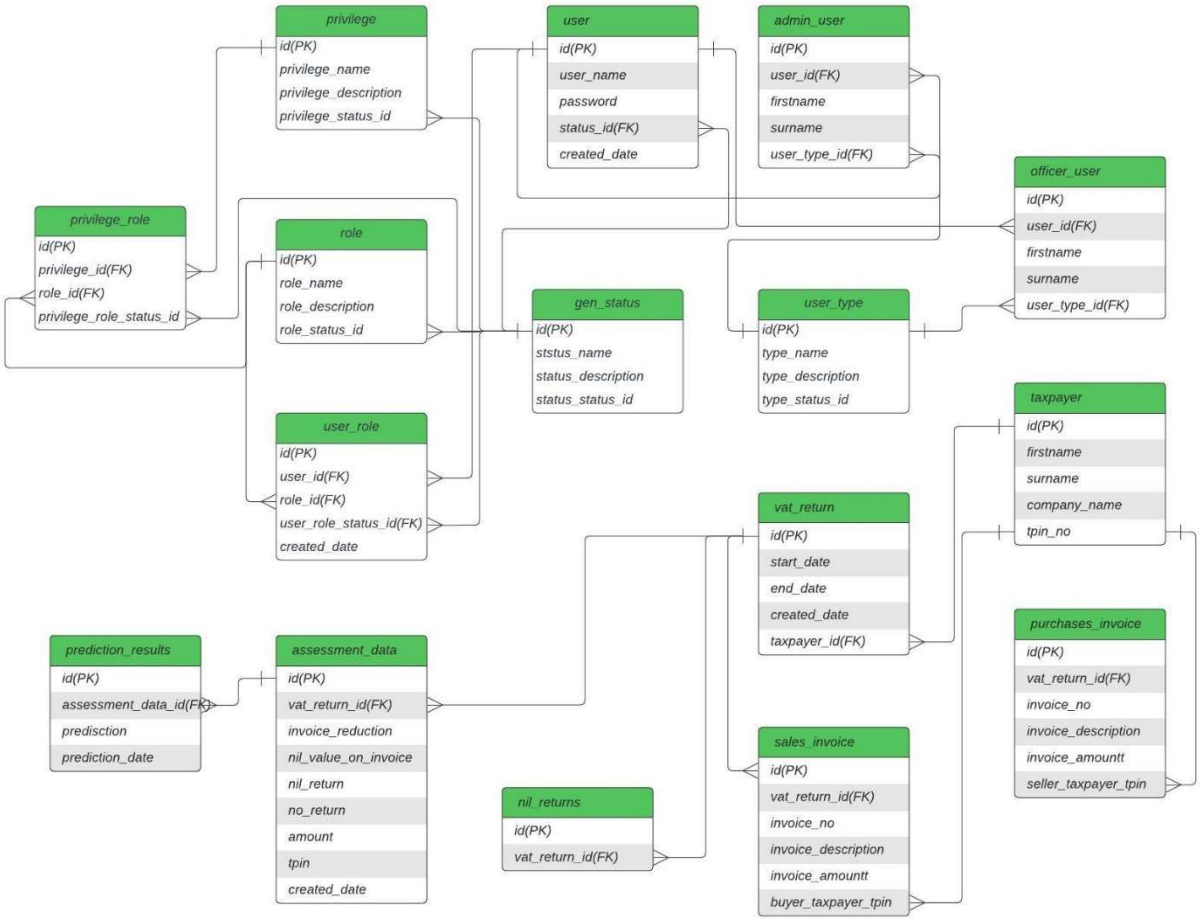


Figure 15: Database Entity relationship diagram

3.4.3.1.4 Database procedures

The data used by the prediction model was designed to be collected from the assessment_data table. This table contained the collection of the consolidated and filtered data from the various tables in the database. A RDBMS procedure was created to contain and execute the logic required to calculate and pull data from other tables into the assessment_data table. The database procedure was used to encapsulate the complex logic required to prepare the data into a single executable unit. This made it easier to maintain the logic and decrease the number of calls to the database to enhance the application's performance when required to execute. At a configurable time interval, using and database job, the procedure runs and inserts new records into the table in readiness for the prediction engine. The fig. 16 below shows an excerpt of the DB procedure from the MySQL database.

```
BEGIN

#Insert records from returns tables
#Invoice reduction and nil value on invoice
insert into assessment_data (vat_return_id, invoice_reduction, nil_value_on_invoice, nil_return, no_return, amount, tpin, start_date, end_date) (
SELECT sell.vat_return_id as 'sell_vat_return_id',
if((pur.invoice_amount - sell.invoice_amount)>0,'1','0') as inv_red, if((pur.invoice_amount - sell.invoice_amount)>0 AND ((sell.invoice_amount) =0),'1','0') as nill_val_on_invoice,
0,0, (pur.invoice_amount - sell.invoice_amount) as "diff", txp.tpin_no, vat_ret.start_date, vat_ret.end_date FROM sales_invoice sell, purchases_invoice pur, vat_returns vat_ret, taxpayer_user txp
WHERE sell.invoice_no = pur.invoice_no
and sell.vat_return_id = vat_ret.id
and vat_ret.taxpayer_id = txp.id
);


#Nil returns
insert into assessment_data (vat_return_id, invoice_reduction, nil_value_on_invoice, nil_return, no_return, amount, tpin, start_date, end_date) (
select 0, 0, 0, 1, 0, a.invoice_amount, a.seller_taxpayer_tpin, vat_ret.start_date, vat_ret.end_date from purchases_invoice a, vat_returns vat_ret
where a.vat_return_id = vat_ret.id
and a.seller_taxpayer_tpin in (
select d.tpin_no from nil_returns b, vat_returns c, taxpayer_user d
where b.return_id = c.id
and c.taxpayer_id = d.id
)
);


#No return
insert into assessment_data (vat_return_id, invoice_reduction, nil_value_on_invoice, nil_return, no_return, amount, tpin, start_date, end_date) (
select 0, 0, 0, 0, 1, pur_inv.invoice_amount, pur_inv.seller_taxpayer_tpin, vat_ret.start_date, vat_ret.end_date from purchases_invoice pur_inv, vat_returns vat_ret
where pur_inv.vat_return_id = vat_ret.id
and pur_inv.seller_taxpayer_tpin not in (select tu.tpin_no  from vat_returns vr, taxpayer_user tu where vr.taxpayer_id = tu.id)
);

END
```

Figure 16: Excerpt from SQL showing data consolidation DB procedure

3.4.3.1.5 UML Activity diagram

An activity diagram was designed and developed for the system using the unified model language UML notation. The activity diagram shows how the system processes flow from one process to the other. The diagram helps visualise and understand how the activities and the dynamic components interact with each other. The system's activity diagram is shown in the diagram below.

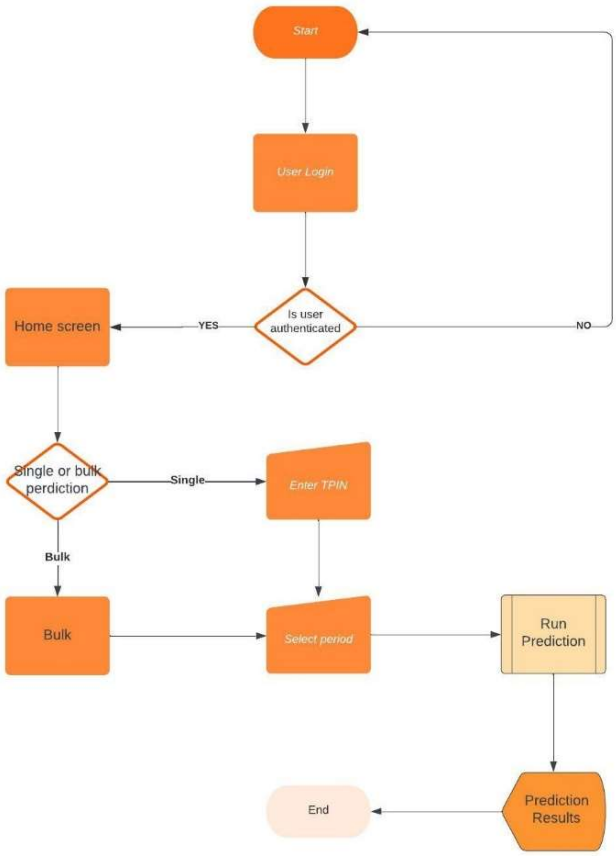


Figure 17: Activity Diagram

Process

- The user attempts to access the system using a login.
- If the user is authenticated successfully, they are redirected to the home screen.
- If the user fails authentication they are redirected back to the login screen.
- The user selects option of single or bulk prediction.
- If the user selected single prediction, they enter the TPIN of the taxpayer whom they would like to run the prediction on and enter the period for the prediction.
- If they user selects bulk, they enter the period for the prediction.
- The prediction engine runs the prediction and displays the results on the screen.

3.4.3.1.6 UML Sequence control diagram

A sequence control diagram was designed and developed for the system using the unified model language UML notation. The sequence control diagram illustrates the interactions among objects in the application and shows the order in which the interactions may occur. Figure 18 below shows this interaction.

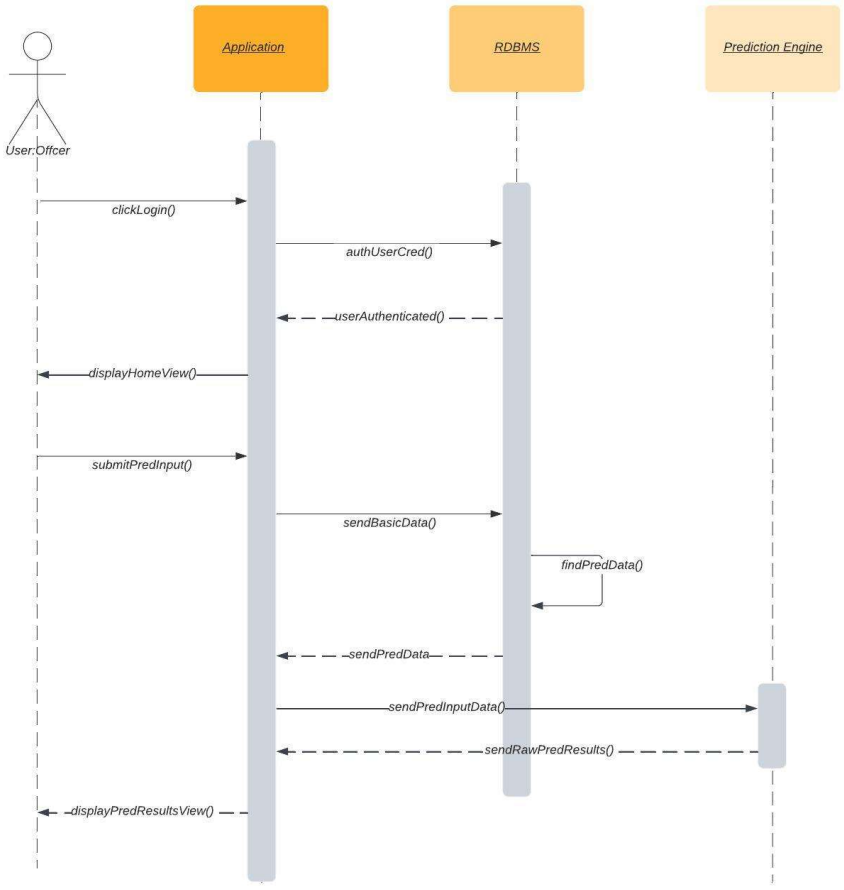


Figure 18: Sequence Control Diagram

Preconditions

- User must already exist on the system
- User must have the required roles and privileges to access the system

Process

- The user interacts with the system by entering the login credentials and clicks submit.
- The application sends the encrypted user credentials for query on the database.
- The database responds to the application with the results.
- The application decrypts the results from the database, authenticates the user and displays the home page to the user.
- The user enters and submits input data for prediction
- The application prepares and sends input data to the database
- Database queries for declarations information based on the data received
- Database responds with the prediction data based on the data it received.
- The application prepared the prediction data and invokes the prediction model with the data.
- The prediction model makes the forecast and provides the application with the outcomes.
- The application prepares the data and displays the results in the appropriate format to the user.

3.4.4 System Implementation

The system prototype was developed in three stages. The first stage involved the process of the development and evaluation of the prediction model. The second stage was the development of the web application and web service that interfaced with users and other systems To predict outcomes about input data from users in production. The second stage involved the development and setup of the application web-hosting environment on which the web application was hosted.

3.4.4.1 Tools and technologies

A number of tools and technologies were employed to achieve some of the sub activities which were integrated into the full system. The following were the technologies that were used to develop the system.

3.4.4.1.1 Technologies

3.4.4.1.1.1 Python

Python is an object-oriented, interpreted, high-level scripting language. It is interactive. Python allows for the programming paradigm known as "Object-Oriented," which encapsulates code inside objects. [92]. It can be integrated with other programing languages. It is easy to learn and write. It is normally used in data analysis, machine learning, artificial intelligence and automation projects. We used python language do develop our machine learning prediction model which was later integrated into our application prototype. The model accepts input data, makes a prediction and produces output. Python allows for the importation of a wide variety of libraries required to perform certain tasks. Libraries in computer programming are a collection of pre-complied programming code, which contain code to perform some specific tasks. The developer imports and makes use of the libraries relevant to their problem thereby avoiding re-inventing the wheel by making use of already developed and tested pieces of code. One of the most common ways to reuse software is through subroutine libraries, especially in the field of scientific programming. [93]. The following are the libraries we used in our python code for the development of the model.

i. **Scikit-learn library**

The most complete open-source machine learning toolkit for Python is called Scikit-learn. Scikit-learn is a collection of efficiently implemented machine learning methods [94]. We used this library the machine learning process by passing the training dataset to the selected machine-learning algorithm. Among the methods contained in the Scikit-learn library is the ensemble package. Learning techniques known as ensemble methods build a group of classifiers and then categorize incoming data points by assigning a (weighted) vote to each one of their predictions [95]. From this package of the library, we used the random forest classifier as the algorithm develop our machine-learning model.

ii. **Pandas library**

A Python module called Pandas offers essential high-level building blocks for using Python to undertake useful, real-world data analysis. [96]. It provides a lot of data processing tools required for the data manipulation as the data is prepared for the machine learning model.

iii. **Matplotlib library**

Matplotlib is a Python library for two-dimensional graphics that can be used for interactive scripting, application development, and the creation of publication-quality images on a variety of operating systems and user interfaces [97]. Understanding your data and gaining insight into the dataset's underlying structure depend on data visualization. These insights assist the scientist in determining which learning algorithm is more appropriate for the particular dataset or in making statistical analysis decisions [98]. In our development process, we used this library to illustrate a number of data visualisation including the model results.

iv. **Seaborn library**

A Python package called Seaborn is used to create statistical visuals. It works closely with pandas data structures and offers a high-level interface to matplotlib [99]. Additionally, Seaborn provides a number of pre-installed themes that users can choose from to alter the way their graphs look [99]. We used seaborn to present our heatmap data presentations to help us analyse our features and their relationships.

v. **Pickle library**

We used pickle library to package and export the model we developed for use in other applications. A Python object structure can be serialized and de-serialized using the pickle module in Python. Pickling allows you to store any Python object on disk. Pickle writes the object to a file after serializing it. A Python object (list, dict, etc.) can be pickled to create a character stream. The notion behind the concept is that all the information needed to reconstruct the object in a different Python script may be found in the character stream [100].

vi. **Flask framework**

We used Flask framework from python to host our web application and web service. Flask is a Python micro framework that offers the fundamental features of a web framework and facilitates the addition of additional plug-ins to expand its functionality and feature set. Flask is referred to as the Python micro framework since it simplifies essential functions while allowing for development extensibility [101]. The web-based flask application's interface and application framework are both provided by Flask [102]. It can be used to host lightweight web applications developed in python.

vii. **Flask_restful**

An addon for Flask called Flask-RESTful makes it possible to quickly construct REST APIs. It is a simple abstraction that is compatible with the current libraries and ORM. Flask-RESTful promotes best practices with little configuration [103]. We used this library to create our RESTful code for our web service.

3.4.4.1.1.2 REST

Representational State Transfer, or REST, is an architectural style that establishes standards for computer systems over the web, facilitating easier system-to-system communication [104]. Clients submit requests to servers to retrieve or modify resources in a REST architecture, and servers respond to these requests [104]. REST allows for systems to exchange data using international standards making it usable with any system developed requiring data exchange. We therefore created a RESTful API which any system can use in order to make use of our trained model for predictions. REST uses four main types of requests, namely GET, POST, PUT and DELETE.

- GET is used to retrieve a specific resource with a specific ID
- POST is used to send data to the server to create or update a resource
- PUT is used to revise a certain resource with a specific identifier
- DELETE is used to remove a certain resource by a specific identifier

3.4.4.1.2 Tools

To put the system's numerous components into practice, we employed a number of tools with different capabilities which allowed to us develop and deploy the system. The following are main tools we used to develop the system. We discuss their strengths and the area they were used in the different areas of the development process.

a) Jupyter notebook

Numerous groups in both industry and science have embraced Jupyter Notebooks extensively. They facilitate the production of literate programming documents that incorporate many forms of rich media together with code, text, and execution results [105]. Jupyter notebook was the main development platform we used to develop the machine-learning model using python and the various python libraries discussed in the section above.

b) Visual Studio Code

We used Visual Studio Code, or VS Code as it is commonly known as the main code editor for our project. Visual Studio Code is a simplified code editor that facilitates development tasks such as version management, task execution, and debugging. It seeks to give developers the precise tools they require for an efficient code-build-debug cycle [106]. It can be used to write code in multiple programing language without the need to switch editor. Some examples of languages supported by VS code include Java, C++, Python and Javascript. We used VS code to write our web application and web service code.

c) Postman

We used postman for testing our REST API web service. The key component of an API developer's toolkit for sharing, testing, documenting, and tracking APIs is Postman. Postman is used by over 3 million engineers and developers globally to create linked software through APIs [107]. Postman allows for accepting of sample input data and connecting to the web service running on a particular URL and port and then obtaining and displaying the result data to the user. It is typically used for testing web services before they are deployed into production for the main systems to consume them.

d) Anaconda navigator

The Anaconda distribution comes with a desktop graphical user interface (GUI) called Anaconda Navigator, which lets you manage conda packages, environments, and channels without the need for command-line instructions [108]. We used anaconda to help us easily manage all the python related applications in our development of the system.

3.4.4.2 System development

The system development was separated into three primary stages. The creation of machine learning models and evaluation was the initial stage. The second phase involved the web application development. Finally, the third phase was the web application testing. Each of the phases was divided into further broken down sub phases. This section of the document discusses these developments in further detail.

3.4.4.2.1 Machine learning Model development

3.4.4.2.1.1 Development Environment setup

The data processing engine was built using random forest, support vector machine and Ada boost machine-learning algorithms using Python. The following is a detailed discussion of the three algorithms. The discussion covers the architecture behind the algorithm, processes, and sub processes that execute when making predictions.

The model development started with the setting up our development language interpreter by installing all the necessary python requirements on our development environment. We then set up anaconda navigator which helped us to set up and manage the other development application required on our project. Using anaconda, we were able to configure python and install our development environment which was jupyter notebook. We used the following commands to setup python using following commands in command prompt.

```
>conda install -c anaconda python
```

Figure 19: Excerpt from cmd prompt python library installation

The following command was used to install jupyter notebook in the command prompt.

```
>conda install -c conda-forge jupyterlab
```

Figure 20: Excerpt from cmd prompt jupyter notepad library installation

Once the environments were installed, we installed all the necessary libraries required for the project. The following were the commands we ran to install all the libraries.

Scikit-learn

```
>conda install -c anaconda scikit-learn
```

Pandas

```
>conda install -c anaconda pandas
```

Matplotlib

```
>conda install -c conda-forge matplotlib
```

Seaborn

```
>conda install -c anaconda seaborn
```

Pickle

```
>conda install -c conda-forge pickle5
```

Figure 21: Excerpt from cmd prompt all library installations

We identified and imported all the necessary libraries required for our project. The screenshot below illustrates our importation code.

Library Importing

```
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
import pandas as pd
import plotly.express as px
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import Binarizer
from sklearn.metrics import roc_curve, roc_auc_score, accuracy_score
from sklearn import ensemble
from sklearn.metrics import confusion_matrix
from sklearn.metrics import r2_score, explained_variance_score, mean_absolute_error
import pickle
```

Figure 22: Excerpt from Jupyter notebook library importation scripts

3.4.4.2.1.2 Data extraction

Once the environment was setup successfully, we extracted the sample-aggregated dataset we extracted from the tax administration system into a CSV file. We declared and created a pandas data frame object and loaded the data into it. The following code was used to load the data from the CSV file.

Loading training and testing dataset

```
In [55]: ast_all_data =  pd.read_csv('DM_DATA/ASSESSMENT_ALL_DATA.csv',sep=",",encoding='cp1252')
```

Figure 23: Excerpt from Jupyter notebook loading training and testing dataset

3.4.4.2.1.3 Data analysis

We performed some data analysis on the data we had imported to get some insights on how the data was structured and observed any cleaning required before feeding it to our machine-learning algorithm. This process is also necessary to ensure that the algorithm only received clean and complete data in order to make accurate predictions. We first inspected the data using code below to show the first ten records.

Inspection of the dataset ¶

```
ast_all_data.head(10)
```

INVOICE_REDUCTION	NIL_VALUE_ON_INVOICE	NIL_DECLARATION	NO_DECLARATION	PAYMENT_COMPLIANCE_SIX_MONTHS	PAYMENT_COMPLIANCE_ONE_YE
N	N	N	Y	Y	
N	N	N	Y	Y	
Y	N	N	N	N	
N	N	N	Y	N	
N	N	N	Y	N	
N	N	N	Y	Y	
N	N	N	Y	Y	
N	N	N	Y	N	
N	N	N	N	N	
N	N	N	Y	Y	

DECLARATION_COMPLIANCE_SIX_MONTHS	DECLARATION_COMPLIANCE_ONE_YEAR	ASSESSMENTS_IN_SIX_MONTHS	FLAG_ASSESSMENT
Y	Y	Y	N
Y	N	N	Y
N	Y	N	N
Y	N	N	N
N	Y	Y	N
Y	N	N	N
N	N	Y	N
Y	Y	Y	N
N	N	N	N
N	N	Y	Y

Figure 24: Excerpt from Jupyter notebook dataset inspection

Using the code below, a count of all the records in the dataset was done which gave us 199,999 records.

```
In [59]: #Total number of records in the training data dataset
         len(ast_all_data)
Out[59]: 199999
```

Figure 25: Excerpt from Jupyter notebook dataset record count

The following code was used to check the dataset for null values. Good practice requires that all null values are identified and filled with average data if the data is unavailable. From the Seaborn heatmap results shown below we were able to determine that we had no null values in our dataset.

```
#Checking for missing values in the dataset

sns.heatmap(ast_all_data.isnull())
```

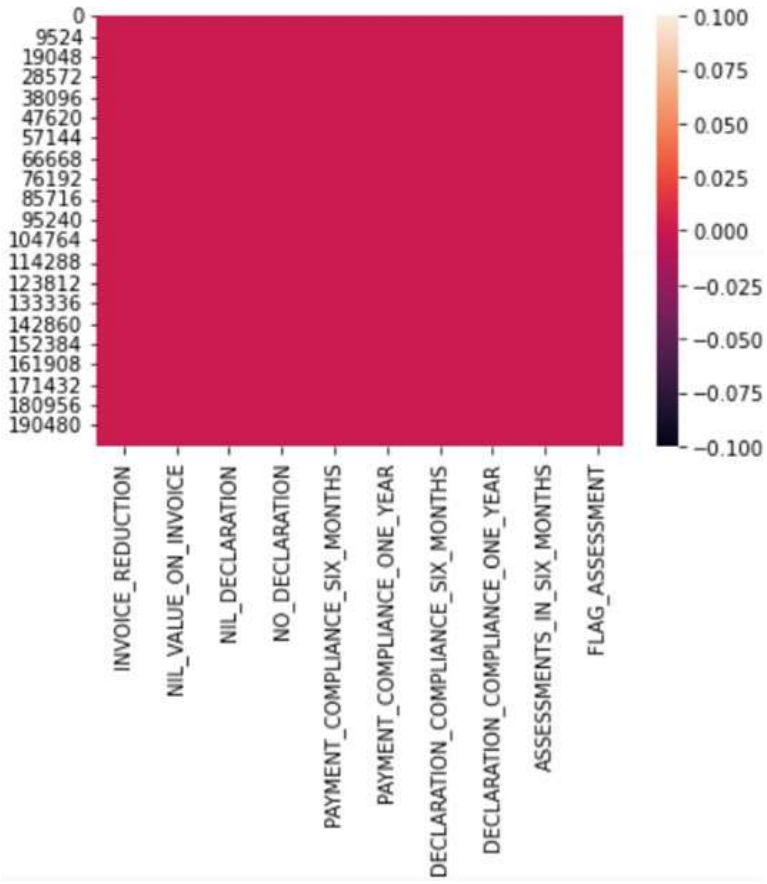


Figure 26: Excerpt from Jupyter notebook checking for missing values in dataset

From our dataset, we were able to do a count on the target variable to see the separation distribution. The target variable in machine learning is the variable of whose value are modelled as a result of the independent variables. We used the code below generate this as shown below:

```
count_flagged_cases = ast_all_data['FLAG_ASSESSMENT'].str.contains('Y').value_counts()[True]

count_not_flagged_cases = ast_all_data['FLAG_ASSESSMENT'].str.contains('Y').value_counts()[False]

print("Total flagged for assessment = ", count_flagged_cases, " | Total not flagged for assessment = ",
```

```
Total flagged for assessment =  76424  | Total not flagged for assessment =  123575
```

Figure 27: Excerpt from Jupyter notebook count of two categories of assessments

We were able to determine that we had a total of 76, 424 of assessments flagged for assessments while 123, 575 were not flagged from a total of 199, 999. We further illustrated this in bar graph as shown below:

```
#Building Bar chart

x = ['Assessment','Non Assessments']
y = [count_flagged_cases, count_not_flagged_cases]

plt.bar(x,y)
plt.title('Comparison of assessment vs non assessment cases')
#plt.legend()
plt.xlabel("Assessment category")
plt.ylabel("No of transactions")
```

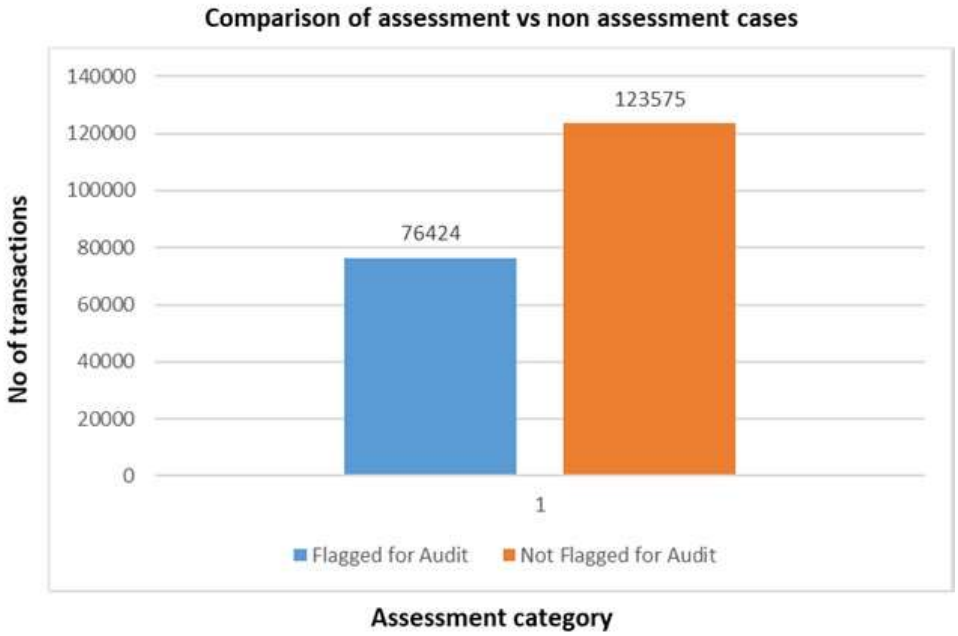


Figure 28: Excerpt from Jupyter notebook comparison of assessment vs non assessments

The data type of the columns in the dataset was inspected using the command show below. This information allowed us to make decisions on which data required conversion in preparation for the machine-learning model.

```
#Check information about data types
ast_all_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 199999 entries, 0 to 199998
Data columns (total 10 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   INVOICE_REDUCTION               199999 non-null  object
 1   NIL_VALUE_ON_INVOICE            199999 non-null  object
 2   NIL_DECLARATION                 199999 non-null  object
 3   NO_DECLARATION                  199999 non-null  object
 4   PAYMENT_COMPLIANCE_SIX_MONTHS   199999 non-null  object
 5   PAYMENT_COMPLIANCE_ONE_YEAR     199999 non-null  object
 6   DECLARATION_COMPLIANCE_SIX_MONTHS  199999 non-null  object
 7   DECLARATION_COMPLIANCE_ONE_YEAR  199999 non-null  object
 8   ASSESSMENTS_IN_SIX_MONTHS       199999 non-null  object
 9   FLAG_ASSESSMENT                 199999 non-null  object
dtypes: object(10)
memory usage: 15.3+ MB
```

Figure 29: Excerpt from Jupyter notebook inspection of data type of sample data

3.4.4.2.1.4 Data preparation

Data preparation involves the converting the data we had into a format that the machine learning engine can use. From the analysis we did we were able to determine that we needed to convert the categorical data to numeric data. The columns INVOICE_REDUCTION, NIL_VALUE_ON_INVOICE, NIL_DECLARATION, NO_DECLARATION, PAYMENT_COMPLIANCE_SIX_MONTHS, PAYMENT_COMPLIANCE_ONE_YEAR, DECLARATION_COMPLIANCE_SIX_MONTHS, DECLARATION_COMPLIANCE_ONE_YEAR, ASSESSMENTS_IN_SIX_MONTHS and FLAG_ASSESSMENT had numerical data represented by the letters "Y" and "N" They were changed to "1" and "0" by us. Y is represented by "1" and N by "0," which is a format suitable for the machine-learning method. We accomplished this by utilizing the Python package One Hot Encoding, which maps categorical data to numerical data. We first created variables for each feature and assigned the label encoder to each of them as shown below:

```
var_invoice_reduction_encoder = LabelEncoder()
var_nil_value_on_invoice_encoder = LabelEncoder()
var_nil_declaration_encoder = LabelEncoder()
var_no_declaration_encoder = LabelEncoder()
var_payment_compliance_six_encoder = LabelEncoder()
var_payment_compliance_one_encoder = LabelEncoder()
var_declaration_compliance_six_encoder = LabelEncoder()
var_declaration_compliance_one_encoder = LabelEncoder()
var_assessments_in_six_encoder = LabelEncoder()
var_flag_assessment_encoder = LabelEncoder()
```

Figure 30: Excerpt from Jupyter notebook creation of new variables

The data was then fit from each of the columns of the data frame into the new variables we created and then applied the fitted label encoder to the pandas data frame columns and shown below as shown below:

```
var_invoice_reduction_encoder.fit(ast_all_data["INVOICE_REDUCTION"])
var_nil_value_on_invoice_encoder.fit(ast_all_data["NIL_VALUE_ON_INVOICE"])
var_nil_declaration_encoder.fit(ast_all_data["NIL_DECLARATION"])
var_no_declaration_encoder.fit(ast_all_data["NO_DECLARATION"])
var_payment_compliance_six_encoder.fit(ast_all_data["PAYMENT_COMPLIANCE_SIX_MONTHS"])
var_payment_compliance_one_encoder.fit(ast_all_data["PAYMENT_COMPLIANCE_ONE_YEAR"])
var_declaration_compliance_six_encoder.fit(ast_all_data["DECLARATION_COMPLIANCE_SIX_MONTHS"])
var_declaration_compliance_one_encoder.fit(ast_all_data["DECLARATION_COMPLIANCE_ONE_YEAR"])
var_assessments_in_six_encoder.fit(ast_all_data["ASSESSMENTS_IN_SIX_MONTHS"])
var_flag_assessment_encoder.fit(ast_all_data["FLAG_ASSESSMENT"])
```

```
var_invoice_reduction_encoder.transform(ast_all_data["INVOICE_REDUCTION"])
var_nil_value_on_invoice_encoder.transform(ast_all_data["NIL_VALUE_ON_INVOICE"])
var_nil_declaration_encoder.transform(ast_all_data["NIL_DECLARATION"])
var_no_declaration_encoder.transform(ast_all_data["NO_DECLARATION"])
var_payment_compliance_six_encoder.transform(ast_all_data["PAYMENT_COMPLIANCE_SIX_MONTHS"])
var_payment_compliance_one_encoder.transform(ast_all_data["PAYMENT_COMPLIANCE_ONE_YEAR"])
var_declaration_compliance_six_encoder.transform(ast_all_data["DECLARATION_COMPLIANCE_SIX_MONTHS"])
var_declaration_compliance_one_encoder.transform(ast_all_data["DECLARATION_COMPLIANCE_ONE_YEAR"])
var_assessments_in_six_encoder.transform(ast_all_data["ASSESSMENTS_IN_SIX_MONTHS"])
var_flag_assessment_encoder.transform(ast_all_data["FLAG_ASSESSMENT"])
```

Figure 31: Excerpt from Jupyter notebook fitting data into new variables

Finally, we created new columns used by the machine learning algorithm and loaded them with the cleaned data. The following code was used to achieve this:

```
ast_all_data["INVOICE_REDUCTION_1"] = var_invoice_reduction_encoder.transform(ast_all_data["INVOICE_REDUCTION"])
ast_all_data["NIL_VALUE_ON_INVOICE_1"] = var_nil_value_on_invoice_encoder.transform(ast_all_data["NIL_VALUE_ON_INVOICE"])
ast_all_data["NIL_DECLARATION_1"] = var_nil_declaration_encoder.transform(ast_all_data["NIL_DECLARATION"])
ast_all_data["NO_DECLARATION_1"] = var_no_declaration_encoder.transform(ast_all_data["NO_DECLARATION"])
ast_all_data["PAYMENT_COMPLIANCE_SIX_MONTHS_1"] = var_payment_compliance_six_encoder.transform(ast_all_data["PAYMENT_COMPLIANCE_S
ast_all_data["PAYMENT_COMPLIANCE_ONE_YEAR_1"] = var_payment_compliance_one_encoder.transform(ast_all_data["PAYMENT_COMPLIANCE_ONE
ast_all_data["DECLARATION_COMPLIANCE_SIX_MONTHS_1"] = var_declaration_compliance_six_encoder.transform(ast_all_data["DECLARATION_
ast_all_data["DECLARATION_COMPLIANCE_ONE_YEAR_1"] = var_declaration_compliance_one_encoder.transform(ast_all_data["DECLARATION_CO
ast_all_data["ASSESSMENTS_IN_SIX_MONTHS_1"] = var_assessments_in_six_encoder.transform(ast_all_data["ASSESSMENTS_IN_SIX_MONTHS"])
ast_all_data["FLAG_ASSESSMENT_1"] = var_flag_assessment_encoder.transform(ast_all_data["FLAG_ASSESSMENT"])
```

Figure 32: Excerpt from Jupyter notebook creation of new columns

The following code was used to inspect the pandas data frame we had created which showed the data in the new columns were created. We also inspected the data type of the new columns we created. It showed that the data was now in int32 data format and was appearing correctly after the data cleaning and preparation process, showing that the one hot encoding process was successful:

```
ast_all_data.head(2)
```

	INVOICE_REDUCTION	NIL_VALUE_ON_INVOICE	NIL_DECLARATION	NO_DECLARATION	PAYMENT_COMPLIANCE_SIX_MONTHS	PAYMENT_COMPLIANCE_ONE
0	N	N	N	Y	Y	
1	N	N	N	Y	Y	

```
ast_all_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 199999 entries, 0 to 199998
Data columns (total 20 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   INVOICE_REDUCTION                     199999 non-null  object
 1   NIL_VALUE_ON_INVOICE                  199999 non-null  object
 2   NIL_DECLARATION                       199999 non-null  object
 3   NO_DECLARATION                        199999 non-null  object
 4   PAYMENT_COMPLIANCE_SIX_MONTHS         199999 non-null  object
 5   PAYMENT_COMPLIANCE_ONE_YEAR           199999 non-null  object
 6   DECLARATION_COMPLIANCE_SIX_MONTHS     199999 non-null  object
 7   DECLARATION_COMPLIANCE_ONE_YEAR       199999 non-null  object
 8   ASSESSMENTS_IN_SIX_MONTHS             199999 non-null  object
 9   FLAG_ASSESSMENT                       199999 non-null  object
 10  INVOICE_REDUCTION_1                   199999 non-null  int32
 11  NIL_VALUE_ON_INVOICE_1                199999 non-null  int32
 12  NIL_DECLARATION_1                     199999 non-null  int32
 13  NO_DECLARATION_1                      199999 non-null  int32
 14  PAYMENT_COMPLIANCE_SIX_MONTHS_1       199999 non-null  int32
 15  PAYMENT_COMPLIANCE_ONE_YEAR_1         199999 non-null  int32
 16  DECLARATION_COMPLIANCE_SIX_MONTHS_1   199999 non-null  int32
 17  DECLARATION_COMPLIANCE_ONE_YEAR_1     199999 non-null  int32
 18  ASSESSMENTS_IN_SIX_MONTHS_1           199999 non-null  int32
 19  FLAG_ASSESSMENT_1                     199999 non-null  int32
dtypes: int32(10), object(10)
memory usage: 22.9+ MB
```

Figure 33: Excerpt from Jupyter notebook inspecting new data types

At this point, the data was prepared and ready for splitting between training and testing set and subsequent input into the machine-learning algorithm.

3.4.4.2.1.5 Model development

The model development process was began by dividing the dataset into testing and training sets. We divided the data into 20% for training and 80% for testing using the train_test_split module of sklearn to accomplish this in Python. The code below was used to perform the train-test-split and inspect the var_x_train variable:

```
var_x_train, var_x_test, var_y_train, var_y_test = train_test_split(ast_all_data[['INVOICE_REDUCTION_1',
```

	INVOICE_REDUCTION_1	NIL_VALUE_ON_INVOICE_1	NIL_DECLARATION_1	NO_DECLARATION_1	PAYMENT_COMPLIANCE_SIX_MONTHS_1
85948	0	0	0	0	0
30124	0	0	0	0	0
117758	0	0	0	0	1
30787	0	0	0	0	0
142734	0	0	0	1	1
...	...	...	...	...	...
68774	0	0	0	0	1
5604	0	0	0	1	1
161140	0	0	0	1	0
87554	0	0	0	1	0
193420	0	0	0	0	0

159999 rows × 9 columns

Figure 34: Excerpt from Jupyter notebook performing train test split

After setting up the random forest classifier, we fed it the test data and set the n_estimators to 200. The number of trees you wish to construct (n_estimators) before utilizing the maximum voting or prediction averages [109]. One of the key variables that can be adjusted to improve the performance of the model is this one. In general, greater results could come from a bigger n_estimator value, but the function will execute more slowly. We then passed the training data to the algorithm using the fit method as shown below:

```
rf_clf = ensemble.RandomForestClassifier(n_estimators=200)

rf_clf.fit(var_x_train, var_y_train.values.ravel())
```

Figure 35: Excerpt from Jupyter notebook training random forest model

The Support Vector Machine and AdaBoost models were also generated and their performance evaluated. The score method which quantifies the proportion of accurate predictions made by the model, or the number of labels it correctly identified out of all the predictions it made [110] along with other model evaluation techniques were performed on the model. We used the confusion matrix, Roc Curve and LogLost for the model evaluation. The results of the evaluations showed that the Random Forest model performed best. The detailed evaluation of the models is discussed further in the following chapter

```
rf_clf.score(var_x_test, var_y_test)
```

Figure 36: Excerpt from Jupyter notebook score method model evaluation

Finally, we packaged and exported the model to the local file system for in preparation for integration with our web application development. The pickle library in python was used for this process.

```
with open('samp_model_rand_forest.pkl', 'wb') as file:pickle.dump(rf_clf, file)
```

Figure 37: Excerpt from Jupyter notebook exporting model using pickle library

3.4.4.2.1.6 Application Code development

The web application was developed in two integrated modules. The first module was developed for users to interact with directly to allow them to make predictions on declarations on demand. In this module, a web user interface is provided for the user to input data they need predictions on and results are displayed on the screen for further action and review. The second module was developed as an application programming interface (API) web service for system integration and processing predictions on large number of samples in real-time on request from the external system. This module works as a back-end sub system and does not have a user interface. Python was used to construct both the web application and the API web service apps, which were hosted on the Flask framework. Visio Studio Code (VS code) was used to develop the applications modules.

The development of the web application logic and functionary was done in python and flask. We used three main python libraries to achieve our development goals. We therefore imported flask, pickle and pandas libraries into our project. We then loaded the machine-learning model we had developed, trained and exported in .pkl format into memory. We used the following code to perform the import of the model:

```
with open(f'C:/Flask/webapp/model/samp_model_rand_forest.pkl', 'rb') as f:
    model = pickle.load(f)
```

Figure 38: Excerpt from VS Code loading model into memory web application module

The GET and POST methods for passing and requesting data from the web server. When the data is processed by the web server, it is formatted prepared and run in the prediction engine. The results of the prediction are formatted and prepared for the web page view. The user can see the prediction's outcomes once the view has been produced for display.

The development of the API started with the importation of flask, pickle and pandas libraries into the project. We then loaded the machine learning model we had developed in .pkl format into memory. We used the following code to perform the import:

```
with open(f'C:/Flask/webapp/model/samp_model_rand_forest.pkl', 'rb') as f:
    model = pickle.load(f)
```

Figure 39: Excerpt from VS Code loading model into memory API module

A method to receive data from the client in JSON format was developed. Language-neutral, text-based JSON is a simple format for exchanging data across languages. [111]. The most widely used format for sending API requests and responses via the HTTP protocol is JSON (JavaScript Oriented Notation), which is a language that is easily understood by both developers and machines [112]. The method receives the data in a POST request and un-marshals the input data for further manipulation. When the data is un-marshalled, it is processed and prepared for input into the prediction model. The prediction along with the original data is then prepared and sent as a response to the client in JSON format.

We used POSTMAN application to test the API. We used it to create a request object in JSON format and passed it in the URL configured for the web service. The screenshot below illustrates the code that was written to handle POST requests and extract the input data from the request.

```
@app.route('/checktransactionspost', methods=['POST'])
def json_function1():
    request_data = request.get_json()

    invoiceReduction = None
    nilValOnInvoice = None
    nilReturn = None
    noReturn = None
    paymentComplianceSixMonths = None
    paymentComplanceOneYear = None
    returnComplianceSixMonths = None
    returnComplianceOneYear = None
    assessmentsInSixMonths = None
    isFlagedForAssessment = None


    invoiceReduction = request_data['invoiceReduction']
    nilValOnInvoice = request_data['nilValOnInvoice']
    nilReturn = request_data['nilReturn']
    noReturn = request_data['noReturn']
    paymentComplianceSixMonths = request_data['paymentComplianceSixMonths']
    paymentComplanceOneYear = request_data['paymentComplanceOneYear']
    returnComplianceSixMonths = request_data['returnComplianceSixMonths']
    returnComplianceOneYear = request_data['returnComplianceOneYear']
    assessmentsInSixMonths = request_data['assessmentsInSixMonths']
```

Figure 40: Excerpt from VS Code marshalling and unmarshalling of data

3.4.4.3 Security Implementation

The security implementation was developed to ensure integrity of system and the data processed by the system. The system uses standard best practice solutions to protect users logged into the system and their data as it is transported to other external systems. We proposed a solution that would provide security at both the application and network layer. The implementation of the security on the network layer was out of scope of this research. We however provided a proposal that would ensure adequate security on the network layer. We focused on the application layer security in this research. The security was developed at user login and session management which ensured secure login and password encryption. We also implemented encryption of the data as it sent to the third party applications.

a) Application layer Security

Security on the application layer was achieved using secure and efficient data encryption. This encryption was achieved using symmetric encryption to convert the JSON body response into cypher text. This takes advantage of the symmetric encryption speed. When using symmetric cryptography, the sender and the recipient share a key for encryption and decryption [113]. We used the Fernet module under python to achieve symmetric encryption. Fernet is a module in python contained in the cryptography package which generates a unique security key used to encrypt and decrypt data. It uses 128 bit Advanced Encryption Standards (AES) with Cypher Block Chaining to encrypt the data. Fernet uses padding for added security. Padding is the process of adding extra data to ensure the input fits the required size there by improving security levels by hiding the length of the encrypted data [114]. Fernet uses PKCS #7 padding. It achieves authentication through Hash-based Message Authentication Code (HMAC) which uses SHA256 hashing algorithm.

Implementation of encryption of the data started with the generation of the encryption key using Fernet. The generated key was then saved to the database for later use in subsequent encryption and decryption requirements in the application. We developed a separate function to host this implementation on the system. The excerpt below shows how the key was generated from the code and stored in the database.

```
key = Fernet.generate_key()

print("This key is..... : ", key)

f = Fernet(key)
convkey = key.decode()

#Saving key to DB
cursor = mysql.connection.cursor()

sql = "UPDATE enc_keys set key_value = %s"
cursor.execute(sql, [key])
mysql.connection.commit()

print(cursor.rowcount, "enc key record updated.")
```

Figure 41: Excerpt from VSCode showing generation and storage of encryption key

Encryption was implemented at the point in the code after the data was predicted and processed and was ready to be returned to the requesting service via the web service. In order to achieve encryption, we first retrieved the stored key from the database. The output payload was then encrypted and the encrypted version of the payload was returned by service. The excerpt below shows this implementation from the code.

```
#Encrpt payload

#Retrive encryption key from the database
cursor = mysql.connection.cursor()
sql = "SELECT key_value from enc_keys"
cursor.execute(sql)
result_set = cursor.fetchall()
for row in result_set:
        dbKeyValue = row[0]


f = Fernet(dbKeyValue)
encMessage = f.encrypt(b'retVal')
#End of encryption

#Return with encryption
return encMessage

#Return without encrption
#return retVal
```

Figure 42: Excerpt from VSCode showing encryption of the output payload

The implementation allowed for the switching between encrypted and unencrypted modes where required. The figure 43 below shows the output of the encrypted and the unencrypted modes.

Figure 43: Excerpt from Postman showing unencrypted and encrypted output payload

b) Network and Transport layer Security

Under the network and transport layer, the focus was on securing the network infrastructure and communication channels to prevent unauthorized access, attacks, and data breaches. The layer security provides a standard for securing internet communication.

We therefore proposed the implementation of a firewall to monitor and traffic and access to the network and virtual private network to establish secure networks with the third parties connecting to the prediction web service while using public networks. This is achieved by encrypting the communication between the two parties communication. This therefore provides a second encryption in addition to the encryption implemented on the application layer.

3.5 Chapter Summary

In this chapter we looked at the methodology used in the project. We identified the CRISP-DM methodology for data mining projects for use in this project implementation. We discussed the processes and activities that were involved at phase of the cycle. We reviewed the current implementation of the audit process and we proposed solution to the identified gaps in the existing process. In this chapter we designed the system and its architecture from the various aspects of the system which include the database, network setup, front-end design, application and security. We presented how the system was implemented and provided how we leveraged on the available tools in order to achieve the goal.

4. RESULTS

4.1 Introduction

A thorough analysis of the outcomes of our modeling, testing, experimentation, and tuning is given in this part. To see relationships between and among variables, we employed the correlation heatmap. We built our model based on the random forest machine-learning model. We tuned the model parameters in order to get the best possible parameter setup by running a series of iterations of epochs. We evaluated the model performance using tools such the confusion matrix and ROC curve. These tools were able to show us how the model performed in terms of accuracy and precision.

4.2 Correlations

The correlation heatmap is a data visualisation library that helps us see relationships between numerical variable. It shows the strength and weaknesses of relationships of the variables utilized throughout the model's training and testing. We generated a seaborn correlation heatmap to give us further insight into our data in terms of how the different columns related to each other. The diagram below shows the correlation heat map for our dataset.

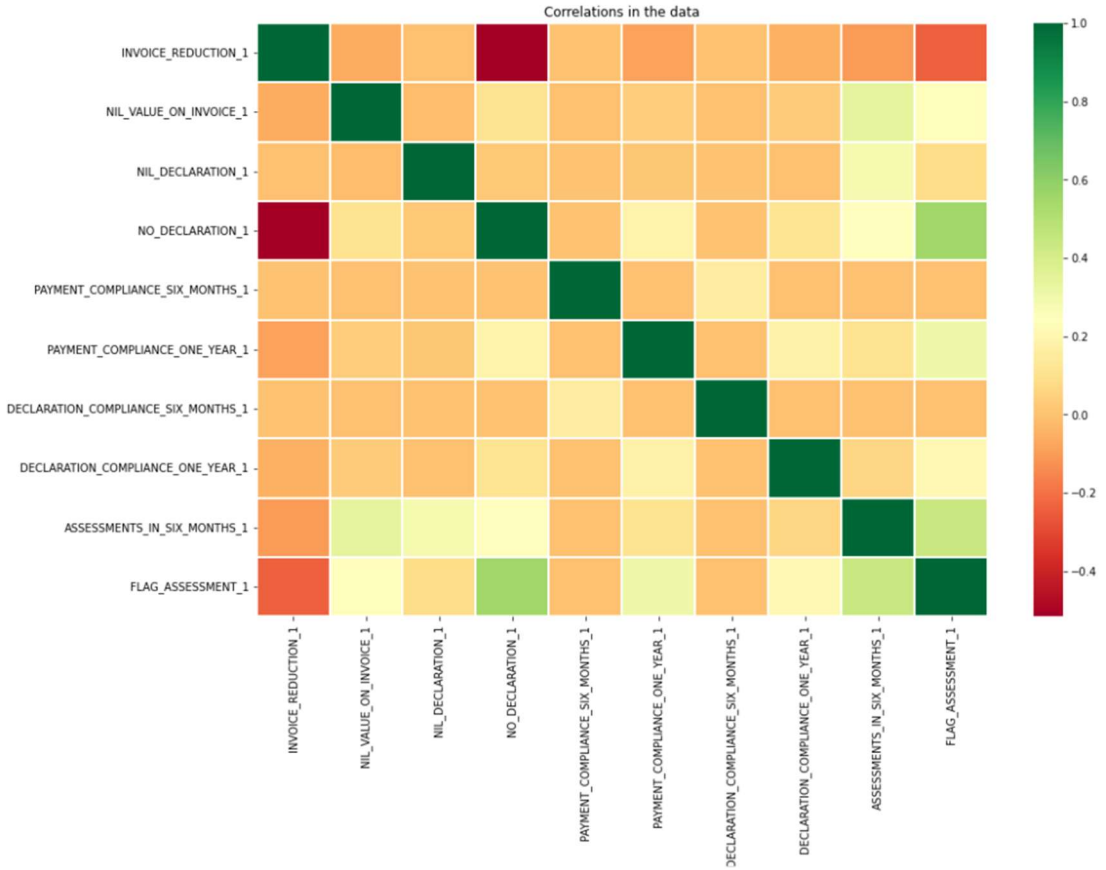


Figure 44: Showing the Correlation heatmap

Table 28: Table showing feature correlations

	INV_RED	NIL_VAL_INV	NIL_DEC	NO_DEC	P_COMP_6_MON	PAY_COMP_1_YR	DEC_COMP_6_MON	DEC_COMP_1_YR	ASST_6_MON	FLAG_ASST
INV_RED	1	-0.063137	-0.002559	-0.516247	-0.000236	-0.084925	-0.000547	-0.049399	-0.10539	-0.243464
NIL_VAL_INV	-0.063137	1	-0.016919	0.111319	-0.001183	0.03639	-0.002201	0.031813	0.34188	0.233449
NIL_DEC	-0.002559	-0.016919	1	0.020561	0.0008	0.015039	-0.000722	-0.002168	0.286606	0.083813
NO_DEC	-0.516247	0.111319	0.020561	1	-0.000222	0.184642	-0.000481	0.114036	0.238209	0.553688
P_COMP_6_MON	-0.000236	-0.001183	0.0008	-0.000222	1	-0.004104	0.156837	-0.002668	-0.001705	-0.00472
PAY_COMP_1_YR	-0.084925	0.03639	0.015039	0.184642	-0.004104	1	-0.004205	0.17556	0.10711	0.305138
DEC_COMP_6_MON	-0.000547	-0.002201	-0.000722	-0.000481	0.156837	-0.004205	1	-0.000975	-0.001321	-0.003572
DEC_COMP_1_YR	-0.049399	0.031813	-0.002168	0.114036	-0.002668	0.17556	-0.000975	1	0.059394	0.208047
ASST_6_MON	-0.10539	0.34188	0.286606	0.238209	-0.001705	0.10711	-0.001321	0.059394	1	0.43812
FLAG_ASST	-0.243464	0.233449	0.083813	0.553688	-0.00472	0.305138	-0.003572	0.208047	0.43812	1

From the correction heatmap, we were able to observe a significant correlation between the instances highlighted for assessment and the no declaration transactions. This finding may suggest that taxpayers who fail to file their declarations on a regular basis are frequently audited and evaluated. A same association may also be observed in the tendency for taxpayers who had previously undergone assessment to undergo auditing and assessment once more. This may be a sign that taxpayers who underreport during an audit and assessment are typically repeat offenders. The revenue authority could use this information to look into these trends more thoroughly and implement tactics and policies that encourage taxpayer compliance..

4.3 Classifier Score

The score method is used to measure the accuracy-related performance of the model. The number indicates how many labels the classifier correctly predicted during the training phase. This provides a reasonable initial indicator of our model's performance.

4.3.1 Random Forest Classifier Score

The Random Forest model was evaluated the model using the score method which produced a score of 0.835975. The excerpt below shows how the method was used the result in Jupyter Notebook.

```
rf_clf.score(var_x_test, var_y_test)
0.835975
```

Figure 45: Excerpt from Jupyter notebook random forest score evaluation

4.3.2 AdaBoost Classifier Score

The AdaBoost model was evaluated the model using the score method which produced a score of 0.829875. The excerpt below shows how the method was used the result in Jupyter Notebook.

```
ada_clf.score(var_x_test_np, var_y_test_np)
0.829875
```

Figure 46: Excerpt from Jupyter notebook AdaBoost score evaluation

4.3.3 Support Vector Machine Classifier Score

The Support Vector Machine model was evaluated the model using the score method which produced a score of 0.81555. The excerpt below shows how the method was used the result in Jupyter Notebook.

```
svm_classifier.score(var_x_test_np, var_y_test_np)
0.81555
```

Figure 47: Excerpt from Jupyter notebook Support Vector Machine score evaluation

4.3.4 Summary of Classifier Scores

The table below displays the outcomes of the scoring method's consolidation comparison for the three classifiers developed.

Table 29: Table showing RF, AdaBoost and SVM score results

Classifier	Score
Random Forest	0.835975
AdaBoost	0.829875
Support Vector Machine	0.81555

4.4 Confusion Matrix Analysis

The confusion matrix was utilized to assess the random forest model's performance even more. An approach that is frequently and widely used to assess a classification problem is the confusion matrix. The confusion matrix allows us to compute statistics, including the number of wrong and right predictions made by the model.

4.4.1 Random Forest Confusion Matrix

The diagram below shows the results the statistics of the confusion matrix for the Random Forest model.

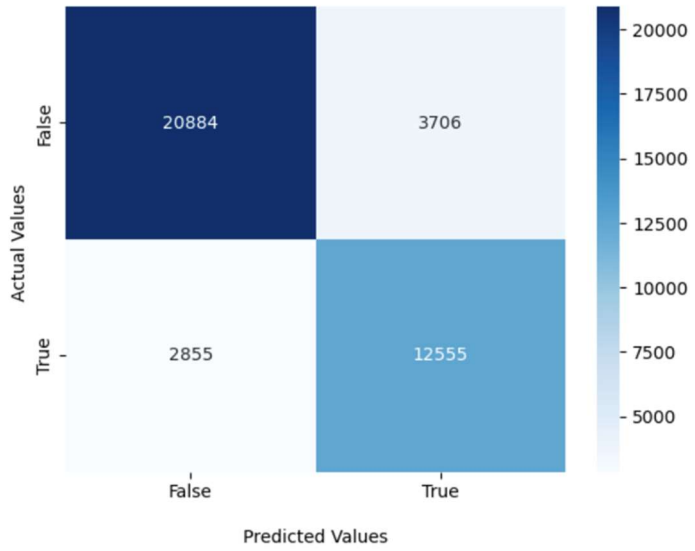


Figure 48: Showing the Random Forest Confusion matrix

From the RF confusion matrix we were able to observe that the model had 20884 true negatives, which is the time the model predicted false correctly. The results showed 12555 true positives, which represents the time the model predicted correctly. The matrix also showed that model had 3706 false positives and 2855 false negatives.

The detailed findings of the confusion matrix examination are displayed in the table below:

Table 30: Table showing Random Forest confusion matrix detailed evaluation results

Classification rate	Description	Calculation	Rate
Accuracy	Demonstrates the total frequency of the classifier's accuracy	(TP+TN)/total = (12555+ 20884)/40000	**0.84**
Precision	Demonstrates the frequency of accuracy when it predicts yes	TP/predicted yes = 12555/16261	**0.77**
Sensitivity	Demonstrates how frequently it guesses "yes" when "yes" is the true answer.	TP/actual yes = 12555/15410	**0.81**
False Positive Rate	Demonstrates the frequency with which it predicts yes when it is actually no	FP/actual no = 3706/24590	**0.15**
Specificity	Demonstrates how frequently it says "no" when it really means "no."	TN/actual no = 20884/24590	**0.85**
Error Rate	Demonstrates to us how frequently the result is incorrect overall.	(FP+FN)/total = (3706+2855)/ 40000	**0.16**

4.4.2 AdaBoost Confusion Matrix

The diagram below shows the results the statistics of the confusion matrix for the AdaBoost model.

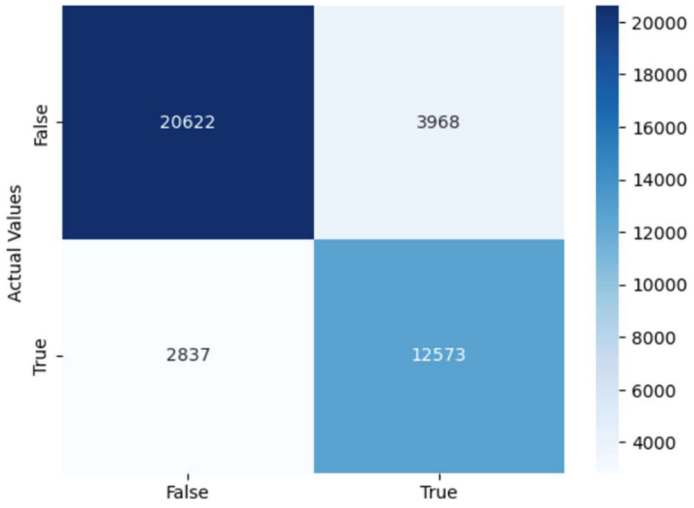


Figure 49: Showing the AdaBoost Confusion matrix

From the AdaBoost confusion matrix we were able to observe that the model had 20622 true negatives, which is the time the model predicted false correctly. The results showed 12573 true positives, which represents the time the model predicted correctly. The matrix also showed that model had 3968 false positives and 2837 false negatives.

The detailed findings of the confusion matrix examination are displayed in the table below:

Table 31: Table showing AdaBoost confusion matrix detailed evaluation results

Classification rate	Description	Calculation	Rate
Accuracy	Demonstrates the total frequency of the classifier's accuracy	(TP+TN)/total = (12573+ 20622)/40000	**0.83**
Precision	Demonstrates the frequency of accuracy when it predicts yes	TP/predicted yes = 12573/16541	**0.76**
Sensitivity	Demonstrates how frequently it guesses "yes" when "yes" is the true answer.	TP/actual yes = 12573/15410	**0.82**
False Positive Rate	Demonstrates the frequency with which it predicts yes when it is actually no	FP/actual no = 3968/24590	**0.16**
Specificity	Demonstrates how frequently it says "no" when it really means "no."	TN/actual no = 20622/24590	**0.84**
Error Rate	Demonstrates to us how frequently the result is incorrect overall.	(FP+FN)/total = (3968+2837)/ 40000	**0.17**

4.4.3 SVM Confusion Matrix

The diagram below shows the results the statistics of the confusion matrix for the Support Vector Machine model.

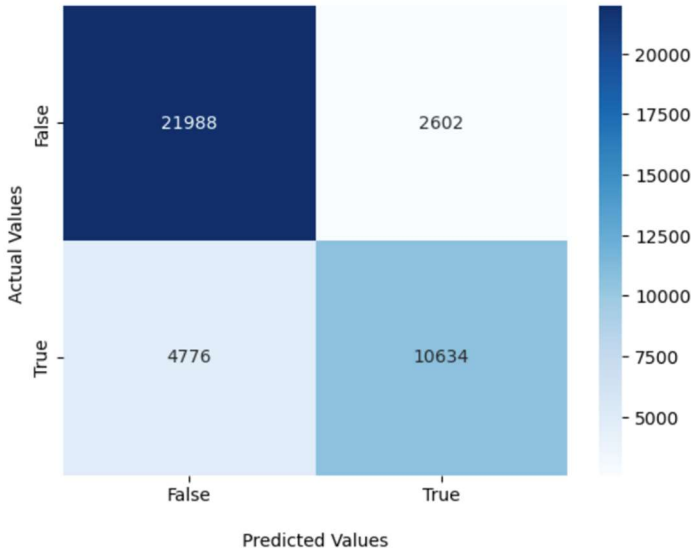


Figure 50: Showing the SVM Confusion matrix

From the SVM confusion matrix we were able to observe that the model had 21988 true negatives, which is the time the model predicted false correctly. The results showed 10634 true positives, which represents the time the model predicted correctly. The matrix also showed that model had 4776 false positives and 2602 false negatives.

The confusion matrix evaluation's detailed results are displayed in the table below:

Table 32: Table showing SVM confusion matrix detailed evaluation results

Classification rate	Description	Calculation	Rate
Accuracy	Demonstrates the total frequency of the classifier's accuracy	(TP+TN)/total = (10634+ 21988)/40000	**0.82**
Precision	Demonstrates the frequency of accuracy when it predicts yes	TP/predicted yes = 10634/13236	**0.8**
Sensitivity	Demonstrates how frequently it guesses "yes" when "yes" is the true answer.	TP/actual yes = 10634/15410	**0.69**
False Positive Rate	Demonstrates the frequency with which it predicts yes when it is actually no	FP/actual no = 2602/24590	**0.11**
Specificity	Demonstrates how frequently it says "no" when it really means "no."	TN/actual no = 21988/24590	**0.89**
Error Rate	Demonstrates to us how frequently the result is incorrect overall.	(FP+FN)/total = (2602+4776)/ 40000	**0.18**

4.4.4 Summary of Confusion Matrix results

The table and chart below shows the summary and comparison of the confusion matrix results. The scores of the classification rates on accuracy, precision, sensitivity, false positivity rate, specificity and error rate were averaged and score for each model was calculated as shown. We were able to observe that random forest scored the highest with a score of 4.96 while AdaBoost and Support Vector Machine scored 4.91 each.

Table 33: Table showing summary and comparison confusion matrix result of the RF, ADA and SVM models

Classification rate	RF	ADA	SVM
Accuracy	0.84	0.83	0.82
Precision	0.77	0.76	0.80
Sensitivity	0.81	0.82	0.69
False Positive Rate	0.15	0.16	0.11
Specificity	0.85	0.84	0.89
Error Rate	0.16	0.17	0.18
Overall Score	**4.96**	**4.91**	**4.91**

4.5 ROC Curve Analysis

The characteristics of the receiver's operation is another instrument we employed to gauge our model's performance was the curve. A classification model's performance is gauged at various threshold values via the ROC curve. As a result, it shows the recall or true positive rate versus the false positive rate at different classification criteria.

4.5.1 Random Forest ROC Curve

Below is the ROC curve that the RF model produced.

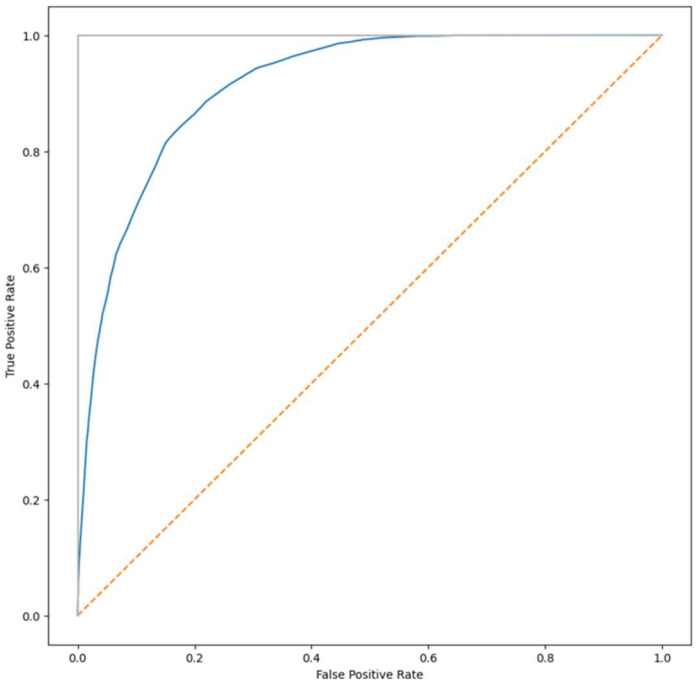


Figure 51: Receiver Operating Characteristic - RF

The AUC demonstrates the model's ability to discriminate between several classes. Our goal was to achieve a high AUC score in every model evaluation. The AUC score for the RF model was 0.9145935971608619. The excerpt below is from our notebook showing the AUC score for the RF Model.

```
print('roc_auc_score for Random Forest: ', roc_auc_score(var_y_test, y_score1_RF))
roc_auc_score for Random Forest:  0.9145935971608619
```

Figure 52: Excerpt from Jupyter notebook AUC score evaluation for the RF model

4.5.2 AdaBoost ROC Curve

The ROC curve produced from the AdaBoost algorithm is shown below.

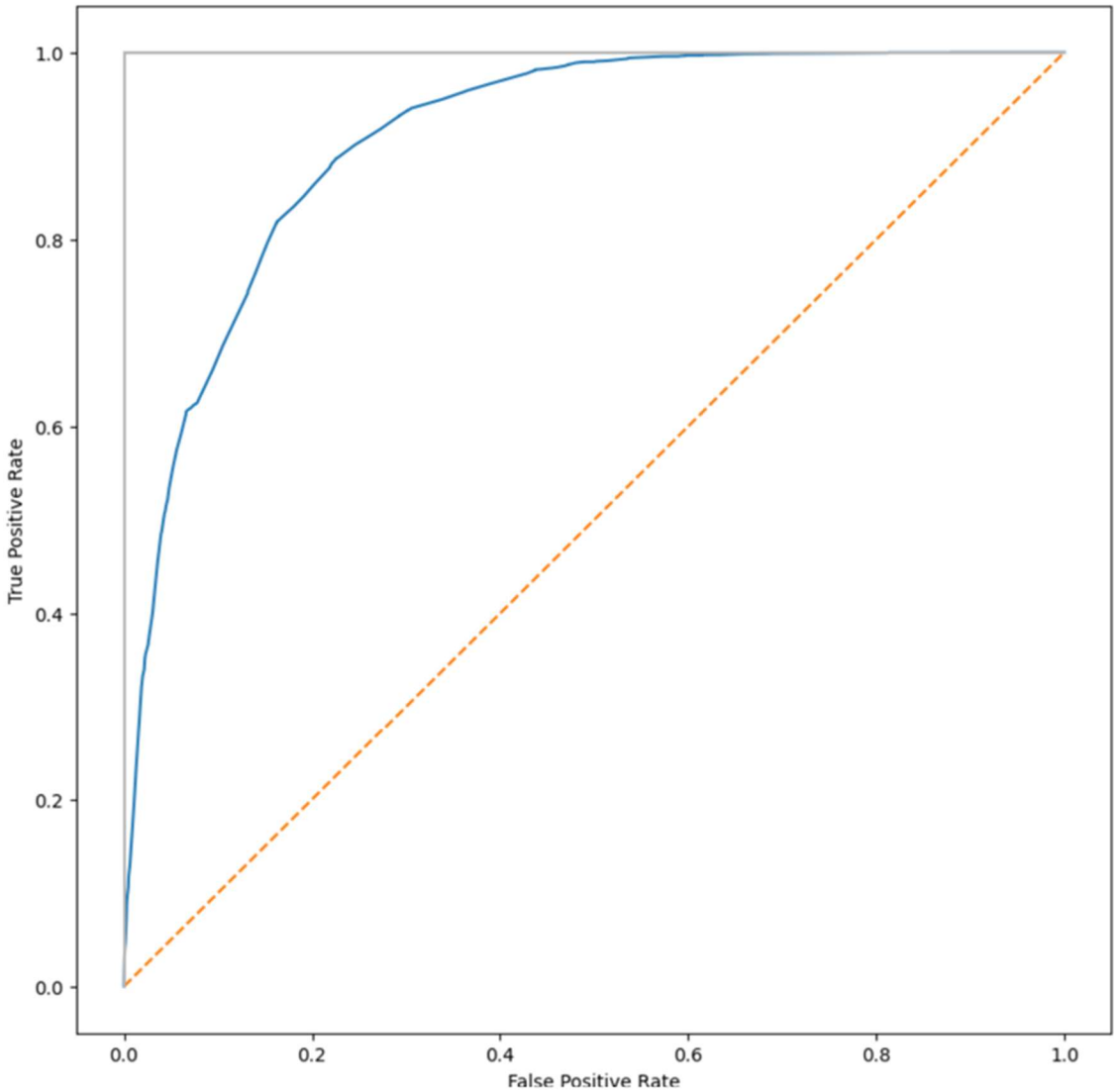


Figure 53: Receiver Operating Characteristic - AdaBoost

The AdaBoost model scored 0.9087725182282094 on the AUC score. The excerpt below is from our notebook showing the AUC score for the AdaBoost Model.

```
print('roc_auc_score for AdaBoost: ', roc_auc_score(var_y_test_np, y_score1_ada))
```

```
roc_auc_score for AdaBoost:  0.9087725182282094
```

Figure 54: Excerpt from Jupyter notebook AUC score evaluation for the AdaBoost model

4.5.3 Support Vector Machine ROC Curve

The ROC curve produced from the SVM algorithm is shown below.

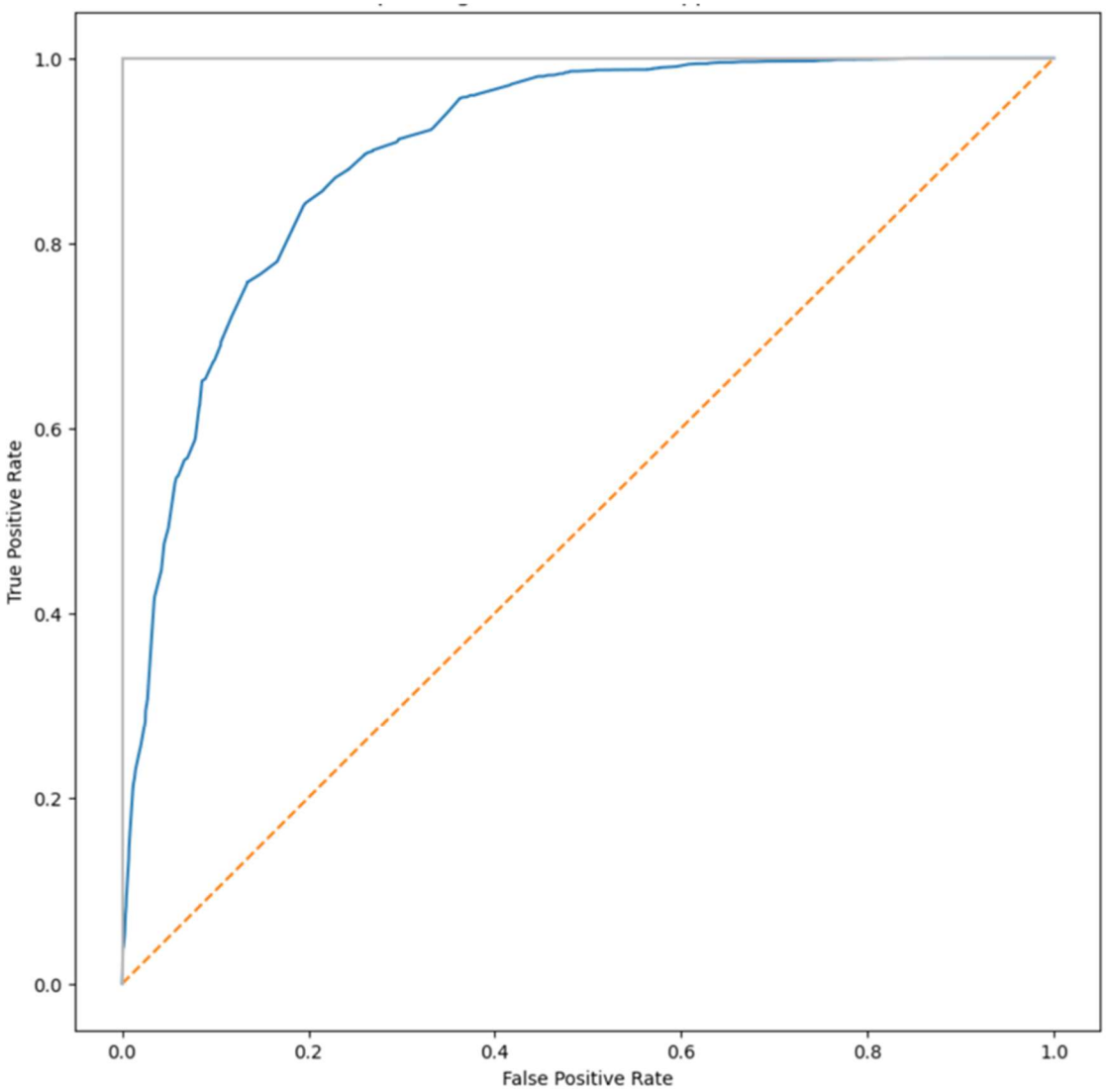


Figure 55: Receiver Operating Characteristic - SVM

The Support Vector Machine model scored 0.8997613687314264 on the AUC score. The excerpt below is from our notebook showing the AUC score for the Support Vector Machine Model.

```
print('roc_auc_score for SVM: ', roc_auc_score(var_y_test_np, y_score1_svm))
```

```
roc_auc_score for SVM:  0.8997613687314264
```

Figure 56: Excerpt from Jupyter notebook AUC score evaluation for the SVM model

4.5.4 Summary of ROC Curve results

The graph below shows the summary and comparison of the ROC curves generated by the three model results.

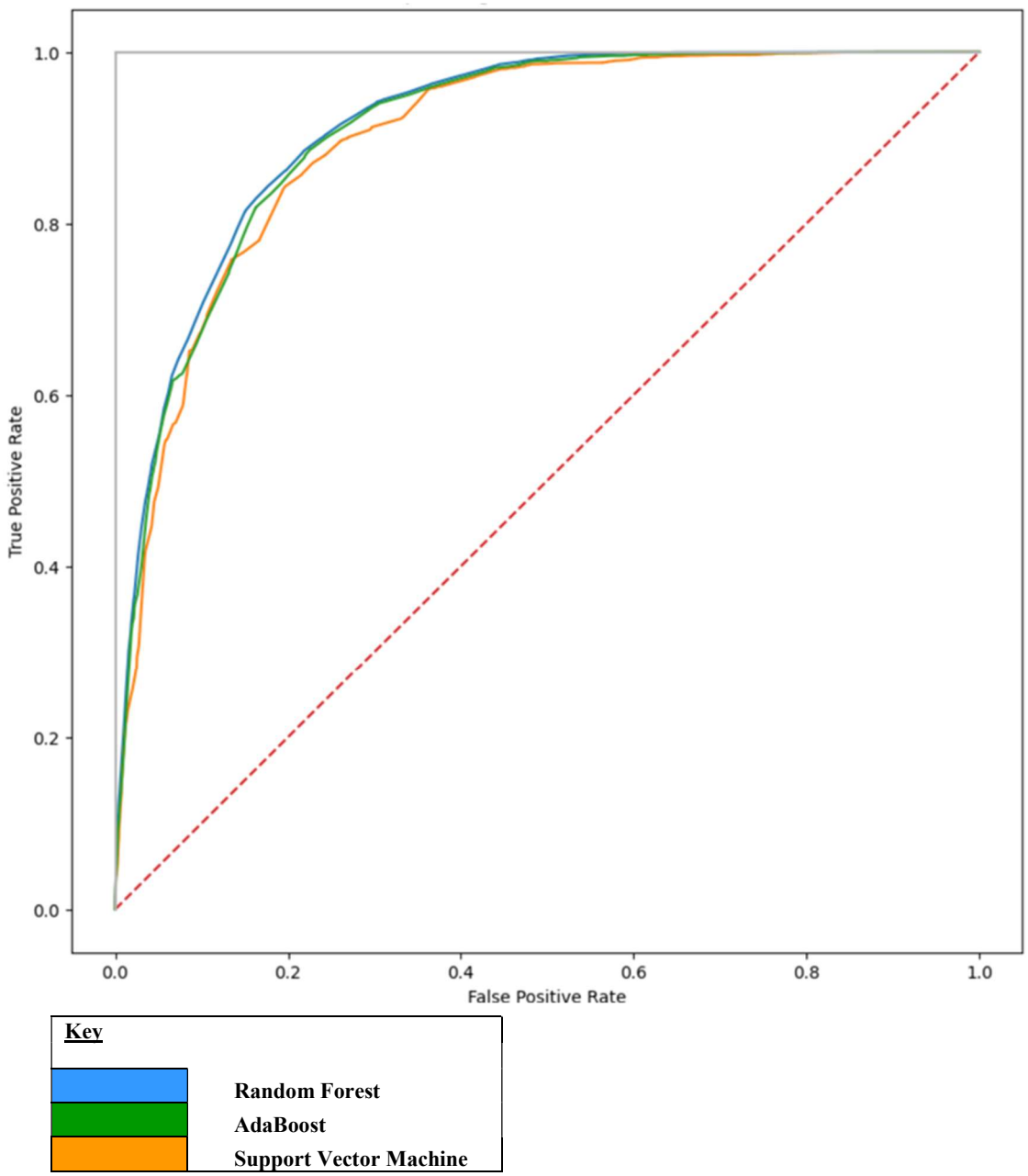


Figure 57: Showing the summary and comparison ROC curve result of the RF, ADA and SVM models

We were able to observe that random forest curve performed the best of the three models. AdaBoost was second followed by Support Vector Machine which had the least score. We also observed that despite AdaBoost and Support Vector Machine performing less than the Random Forest model, they still performed very well as the gap between them and RF was not very big.

The table below shows the summary and comparison of the AUC curves generated by the three model results.

Table 34: Table showing summary and comparison AUC result of the RF, ADA and SVM models

Model	Score
Random Forest	**0.914593597161**
ADA Boost	**0.908772518228**
Support Vector Machine	**0.899761368731**

We were able to observe that Out of the three models, the random forest curve had the best results. Out of the three classifiers, the random forest curve had the best results. AdaBoost was second followed by Support Vector Machine which had the least score. We also observed that despite AdaBoost and Support Vector Machine performing less than the Random Forest model, they still performed very well as the gap between them and RF was relatively minimal.

4.6 Logarithmic Loss (LogLoss) Analysis

When the forecast is a probability between 1 and 0, the performance of a classification model can be measured using logarithmic loss [115]. It calculates the amount that the true label deviates from the expected likelihood. The model performs better when the LogLoss value is closer to zero. The LogLoss results for the three classifiers are as follows.

4.6.1 Random Forest LogLoss

The LogLoss for the random forest model produced a result of 5.667887808845081. This was the best result compared the other two models developed, Support Vector Machine and AdaBoost. The result from the LogLoss produced by the Random Forest gave a fair result.

4.6.2 Support Vector Machine LogLoss

The LogLoss for the support vector machine model produced a result of 6.42512989166967. The Support Vector Machine LogLoss produced the poorest results compared to the other two

models developed, Random Forest and AdaBoost. Despite having the poorest LogLoss result, the margin was not very high compared to the highest

4.6.3 AdaBoost LogLoss

The LogLoss for the AdaBoost model produced a result of 5.856129877308906. The AdaBoost Logloss performed slightly less than the result from the Random Forest model but much better than the Support Vector Machine.

4.6.4 Summary of LogLoss results

The three algorithms' combined LogLoss evaluations are summarized in the table below. Notwithstanding the variations in the outcomes from the three algorithms, the total results demonstrate, the differences were highest difference of 0.8 between the highest and lowest and lowest difference of about 0.2 between the two lowest.

Table 35: Table showing summary and comparison LogLoss results of the RF, ADA and SVM models

Model	Score
Random Forest	**5.667887808845081**
ADA Boost	**5.856129877308906**
Support Vector Machine	**6.42512989166967**

4.7 Web Application results

4.7.1 Web application (module 1)

The web application interface was developed using HTML, CSS and bootstrap. The interface allows for users to interact with, input data and view prediction results. We developed the logic to determine which data was to be displayed to the user based on their input and selections using python code and HTML.

Login page

The login page allows systems users to be authenticated on the system using user names and password stored on the database system. This adds security to the system by only allowing genuine users to access the system. The user password is encrypted as sent to the back-end system for authentication. The figure 39 below shows the login page with an input box for the username and password.

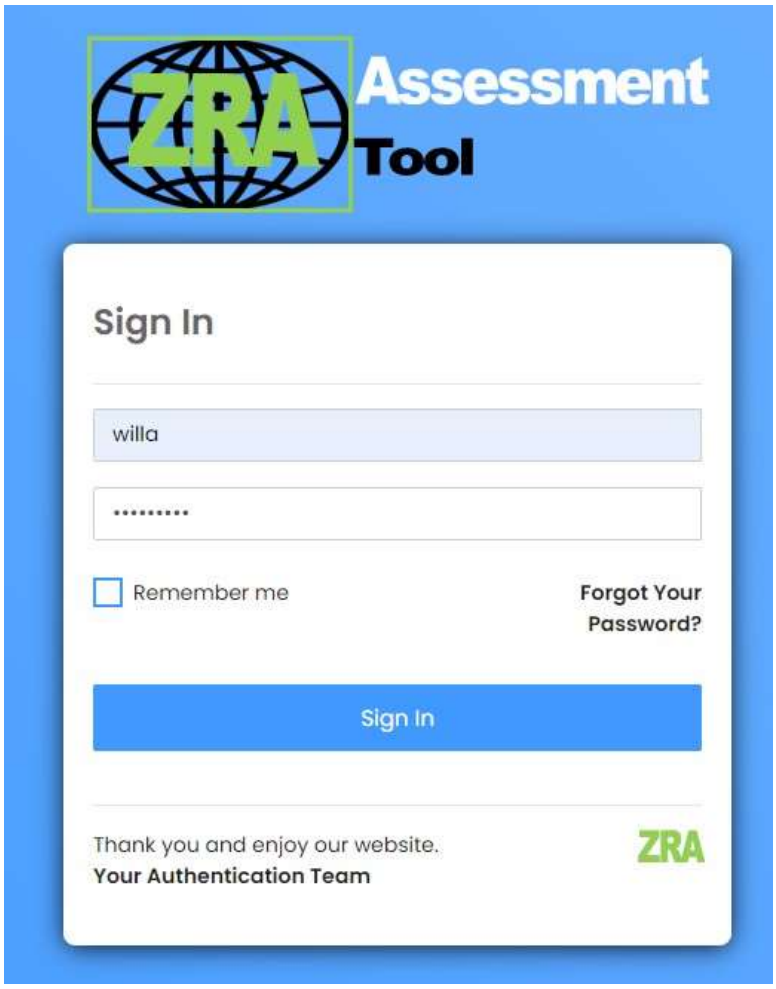


Figure 58: Application login page

Home page

The home page shows general tax statistics generated from the main tax administration system such as compliant tax payers, tax refunds, revenue collection and suspected fraud value and counts. This dashboard will help the tax officer have an overview of current situation with regards to the tax system. It also allows the officer to quickly view and analysis some of the main categories of the tax suppression cases such as invoice reduction, nil values of invoices, nil return filers and non-return filers. It also shows at a glance the high risk cases as results from previous predictions. It allows the user to navigate to the predict single tax payer page and predict bulk taxpayers page. The screenshots below shows the home page with the menu items and general tax statistics from the main tax administration system:

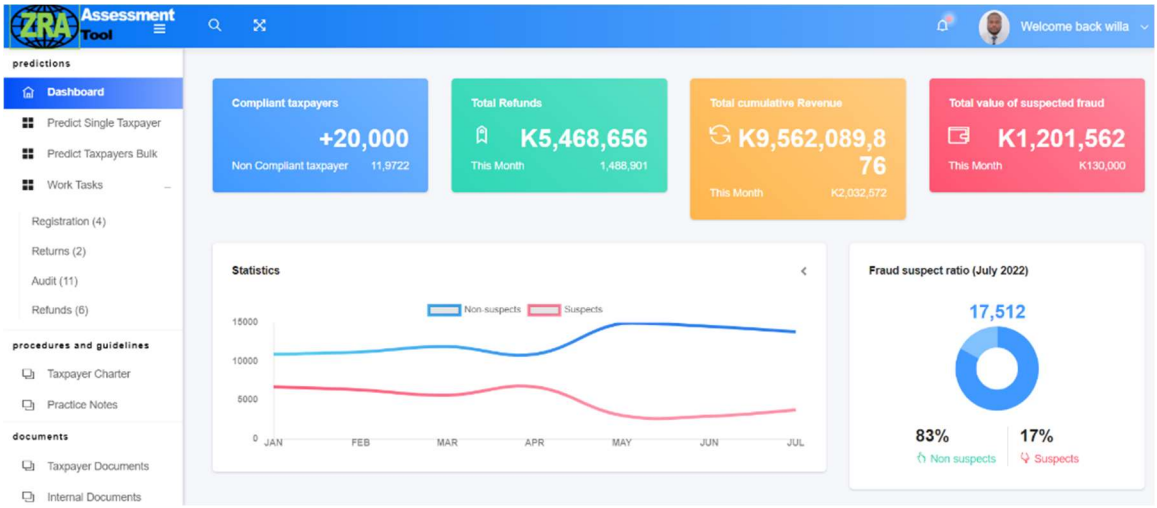


Figure 59: Application home page top

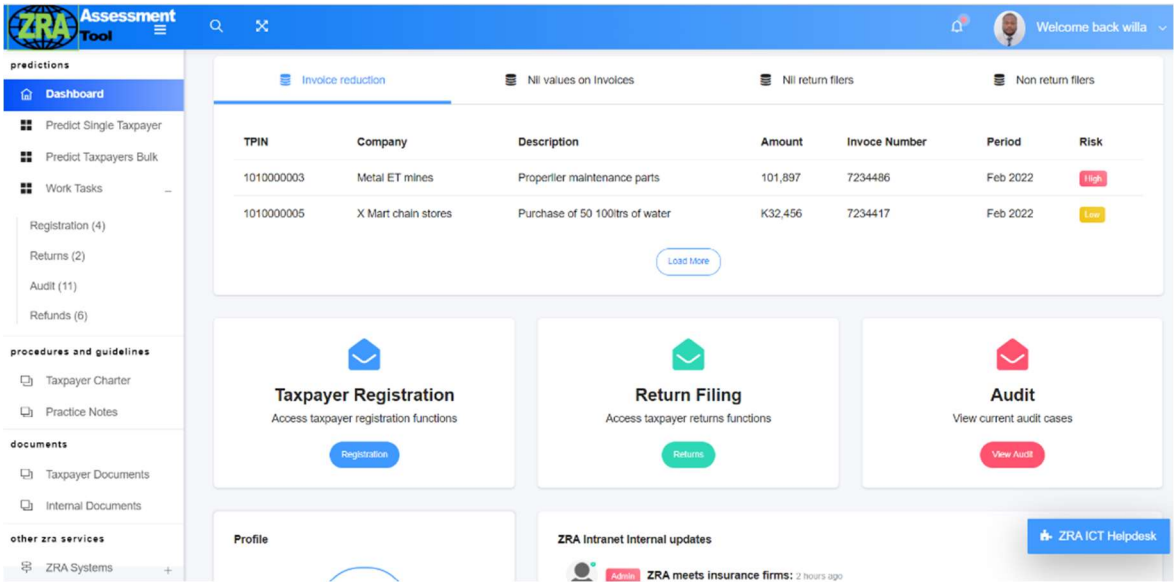


Figure 60: Application home page bottom

Single taxpayer prediction page

The single taxpayer prediction page allows the user to input a taxpayer's TPIN number and allows them to run a prediction on them based on a selected a period and the declarations submitted in that period. The user enters the TPIN and the date the period begins and ends under review and clicks the predict button to start the prediction. Figure 42 below shows a screenshot of the page with the user input.

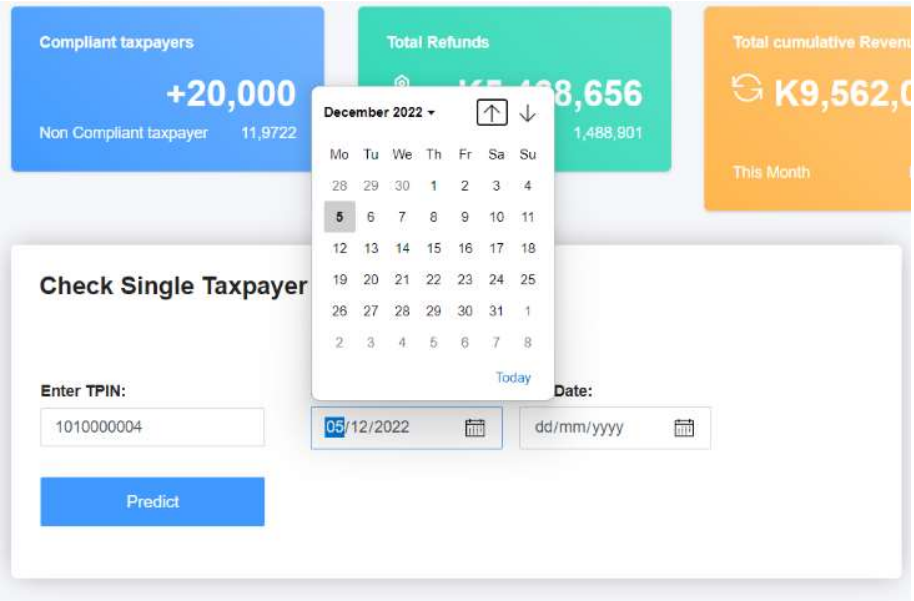


Figure 61: Single taxpayer prediction input page

The prediction for the selected taxpayer and for the period under review is run using the prediction engine in the background and the results are displayed as shown in the figure 43 below.

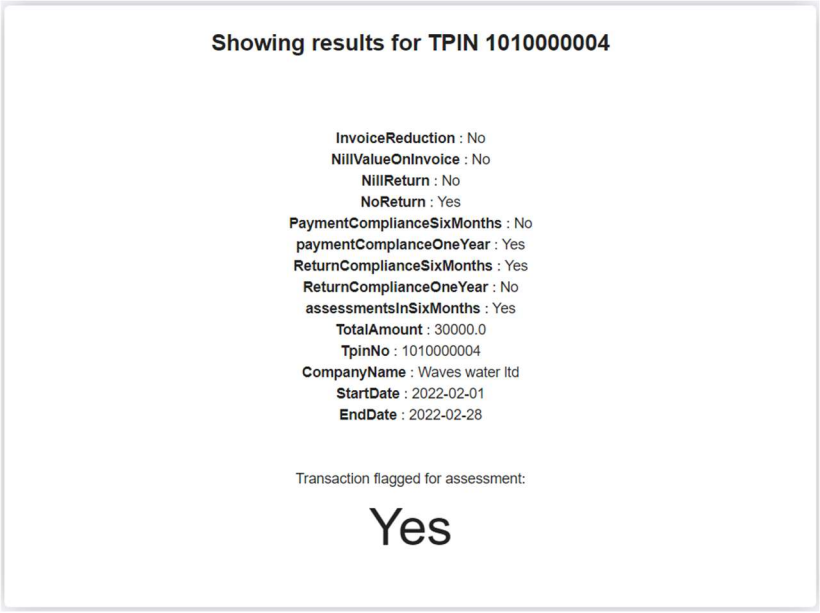


Figure 62: Single taxpayer prediction results page

Bulk taxpayer prediction page

The bulk taxpayer prediction page allows the user to input the start and end period on which to run the prediction based on the declarations submitted in that period. The user enters t the date the period begin and end under review and clicks the predict button to start the prediction. Figure 44 below shows the user input screen.

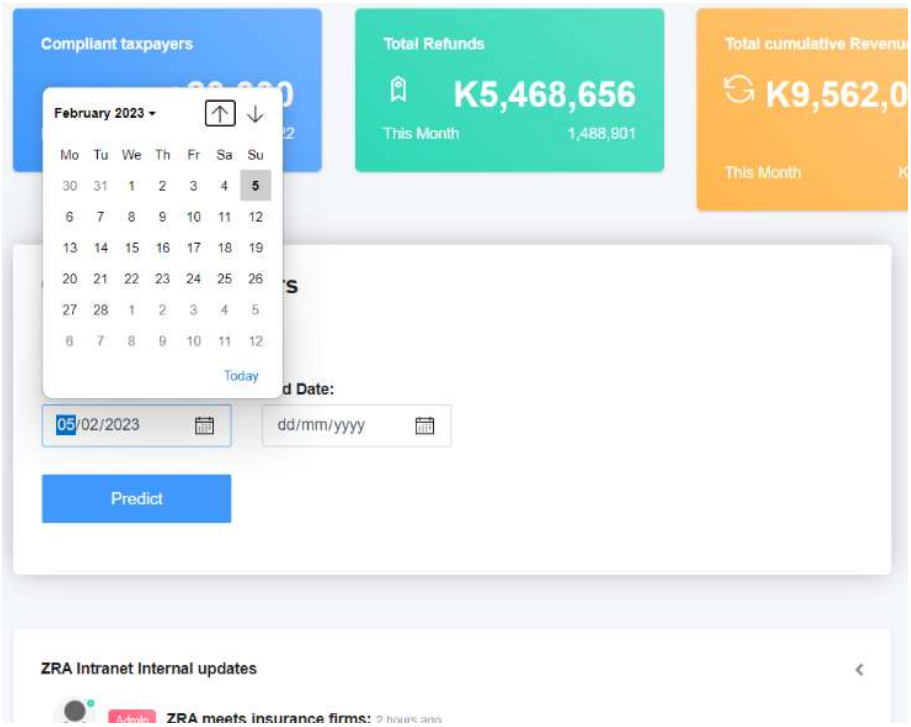
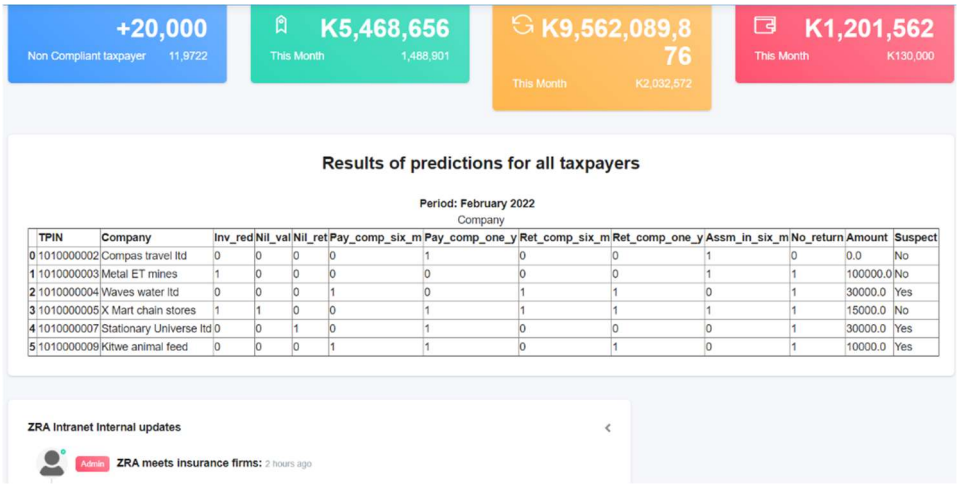


Figure 63: Bulk taxpayer prediction input page

The prediction for the selected period under review is run using the prediction engine in the background and the results are displayed as shown in the figure 45 below.



	TPIN	Company	Inv_red	Nil_val	Nil_ret	Pay_comp_six_m	Pay_comp_one_y	Ret_comp_six_m	Ret_comp_one_y	Assm_in_six_m	No_return	Amount	Suspect
0	1010000002	Compas travel ltd	0	0	0	0	1	0	0	1	0	0.0	No
1	1010000003	Metal ET mines	1	0	0	0	0	0	0	1	1	100000.0	No
2	1010000004	Waves water ltd	0	0	0	1	0	1	1	0	1	30000.0	Yes
3	1010000005	X Mart chain stores	1	1	0	0	1	1	1	1	1	15000.0	No
4	1010000007	Stationary Universe ltd	0	0	1	0	1	0	0	0	1	30000.0	Yes
5	1010000009	Kitwe animal feed	0	0	0	1	1	0	1	0	1	10000.0	Yes

Figure 64: Bulk taxpayer prediction results page

4.7.2 API Web service application (module 2)

The API version of the predication application implementation was designed for use as an API for integration with other external systems. The external systems can implement the prediction

into their existing logic from the response they get from the API. The API also allows other systems to do bulk predictions by sending multiple input data.

The screenshot below shows how the test input data was prepared and sent using POSTMAN application and the response from the API:

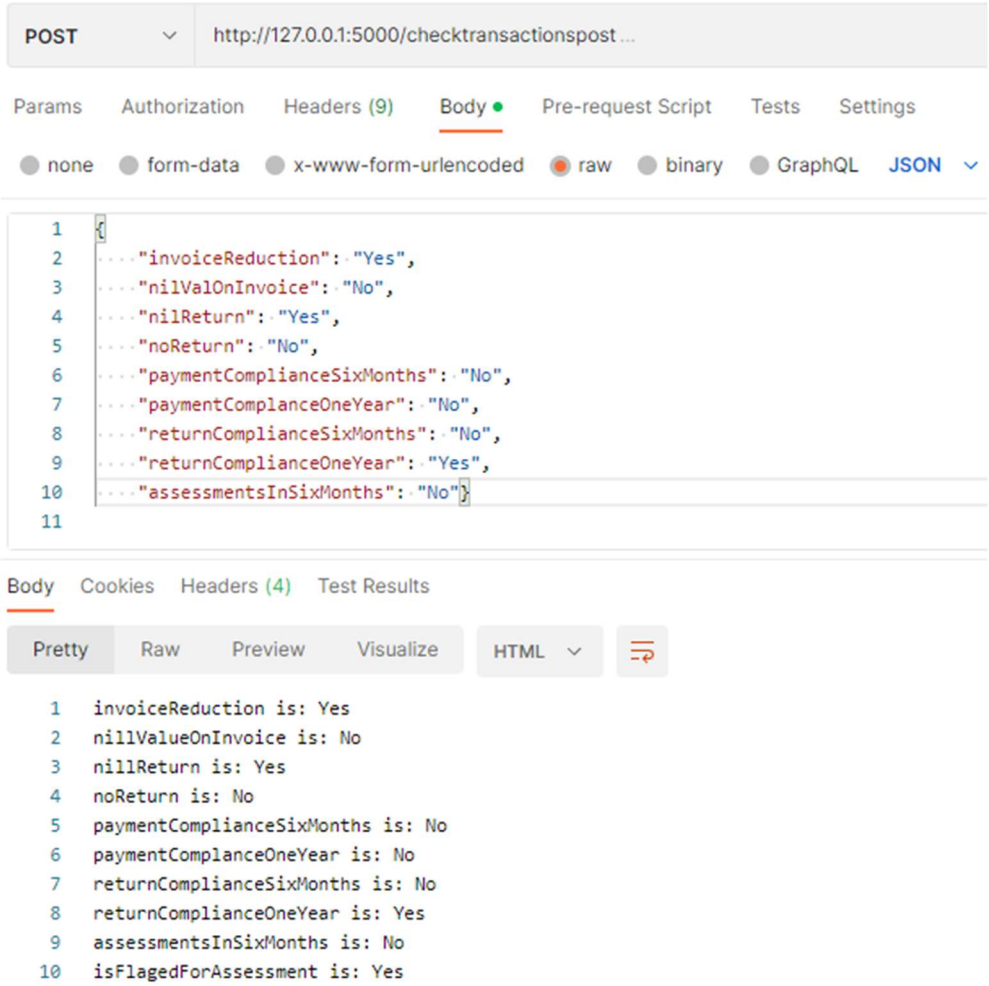


Figure 65: Excerpt from Postman showing sample input data and prediction results

4.8 Chapter Summary

The experiments' findings were reviewed in this chapter and presented in the preceding chapter. We reviewed the performance of the Random Forest, AdaBoost and Support vector machine. We reviewed and compared the results of the three models we developed in order to determine the best performing model. We used the score method model evaluation, confusion matrix and ROC curve and we presented the results from the evaluations. From the evaluations, we observed that the random forest performed best of the three model developed. We also presented the results from the application and API interface showing the prediction results from both modules.

5. DISCUSSION AND CONCLUSION

5.1 Introduction

The research findings are examined in this chapter in light of the first chapter's research questions. Based on the findings of the experiments conducted using the three models covered in the preceding chapter, the chapter offers observations and conclusions. Finally we present some recommendations and proposals for future work in the area of leveraging on machine learning and data mining to improve tax assessments.

5.2 Discussion

In this section, we examine and talk about the consequences of the experiment results for the three constructed models. We also critically analyse the sample data and observe pattern which give us an insight into the taxpayer behaviours in the tax declaration process.

5.2.1 Tax assessment prediction model development

The prerequisite to the first objective which is the development of a tax assessment prediction model using data mining and machine learning, we looked at the first research question which identifies the major challenges affecting tax compliance in Zambia. The research showed that the major challenges affecting tax compliance range from fraud, taxpayer perceptions, poor taxpayer education, poor system controls, ignorance and misconceptions among citizens. The research found that tax evasion is the major causes of poor tax compliance leading to reduced revenue collection. From the trends analysis report published by financial intelligence in 2021 [13], we were able to observe an upward increase trend in the value of tax evasion cases from 2019 to 2021. In 2019 the tax evasion case by value was recorded at K144,000,000 and by 2021, the value had significantly increased to K722,000,000. Other factors such as fraud, corruption and money laundering continued to maintain elevated values with fluctuations during the same period. Figure 66 below shows an illustration of the trend.

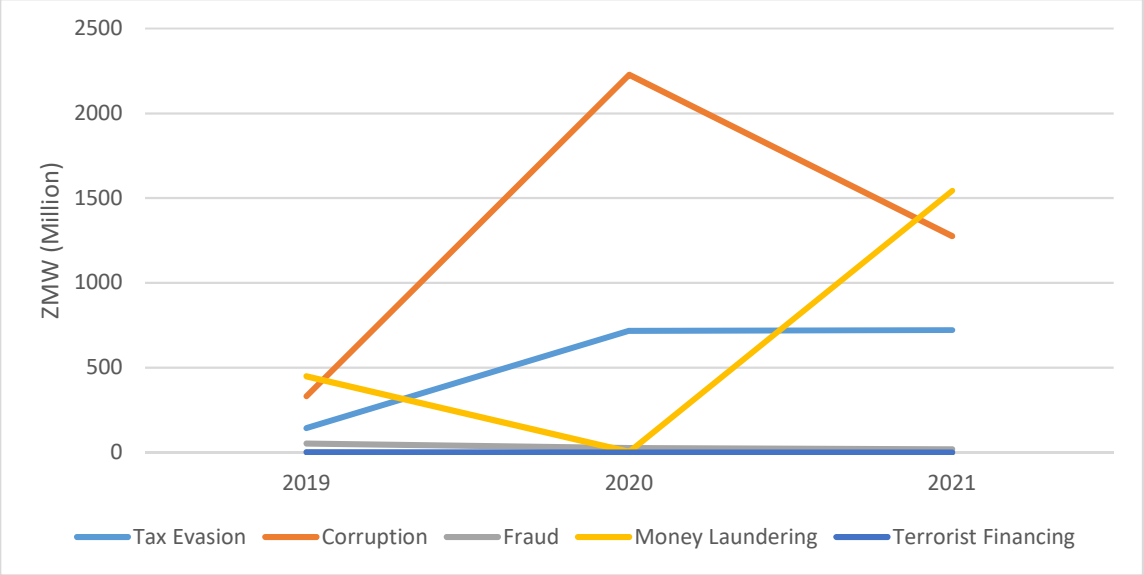


Figure 66 Disseminated Trend Reports by value

According to the Zambia Revenue authority 2022 annual report [48], by the end of the year 2022, the authority had 18 cases were under prosecution while 1 was at planning stage. In that year, the authority recorded 48 civil cases under litigation. 13 were before the Tax Appeals Tribunal, 9 before the Industrial Relations Court, 3 before the Subordinate Court, and 2 before the Supreme Court, 2 before the Court of Appeal, and 1 before the High Court. These figures show that tax evasion and fraud remain a challenge to the revenue authority. In the year 2020, the Zambia Revenue Authority recorded an assessment value of K135.92 million for PAYE, K10.37 million for TOT, K1,724.55 million for Income Tax, K54.76 million for withholding tax, K32.51 million on mineral royalty, K1,674.84 million on VAT and K179.47 million on Excise duty.

We developed the prediction model using past assessment transaction data gathered from the tax administration system. After extracting training and test data from the tax administration system, we created three models with the Random Forest, AdaBoost, and Support Vector Machine methods. The data was split between 20% and 80%. The training and test data consisted of 76, 424 of assessments flagged for assessments and 123, 575 that were not flagged from a total of 199, 999 records. The models we tuned in order to improve the results until there was no further improvement.

5.2.2 Evaluation of tax assessment prediction models

The second objective was to evaluate, compare and identify the prediction model that performs best among the models developed which spoke to research question was on how we could use

data mining and machine learning to identify tax assessments for audit. The three models were evaluated to see the model performed best to be used as the prediction model for the application. We used four techniques for the evaluations of the models. The score method library, confusion matrix, ROC curve and Logarithmic Loss. The score method evaluation according on the findings, the Random Forest model generated the greatest score. of 0.835975 followed by AdaBoost which produced a score of 0.829875. Support Vector Machine performed the least with a score of 0.81555. The results from the Logarithmic Loss evaluation showed that the Random Forest model scored 5.667887808845081 while AdaBoost produced a result of 5.856129877308906 and the Support Vector Machine produced a score of 6.42512989166967. The results from the Logarithmic Loss showed that Random forest performed better than the other two models followed by AdaBoost. The chart below shows the overall model performance comparison of the score method.

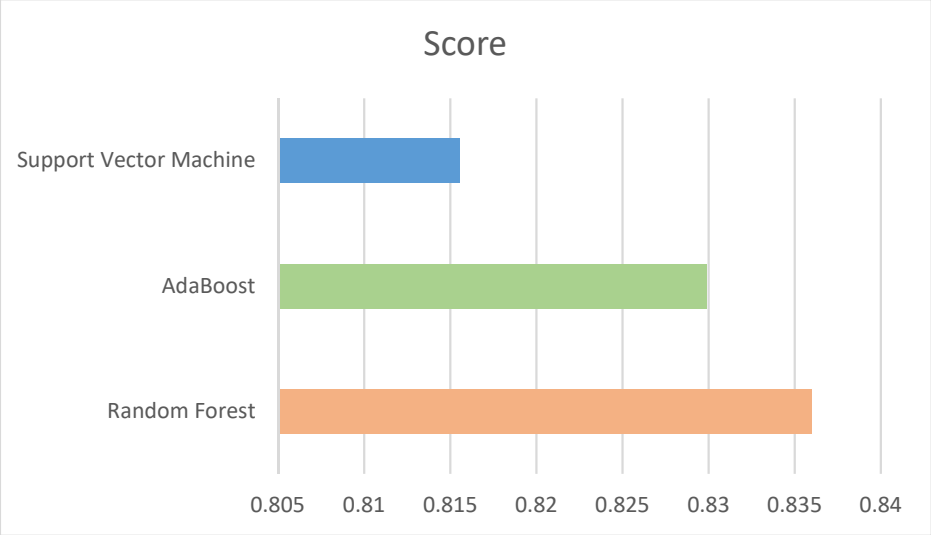


Figure 67: Showing the summary and comparison score method evaluation results of the RF, ADA and SVM models

The Random Forest model's confusion matrix yielded an overall score of 4.96 with accuracy, precision, sensitivity, specificity, false positive rate, and error rate of 84%, 77%, 81%, and 85%, respectively. With an overall score of 4.91, the AdaBoost confusion matrix yielded results of 83% accuracy, 76% precision, 82% sensitivity, 84% specificity, 15% false positive rate, and 17% error rate. With an overall score of 4.91, the Support Vector Machine confusion matrix yielded results of 82% accuracy, 80% precision, 69% sensitivity, 89% specificity, 11% false positive rate, and 18% error rate. With an overall score of 4.96, the Random Forest model's confusion matrix outperformed SVM and AdaBoost, with each scoring 4.91. The confusion matrix's model performance comparison is displayed in the chart below.

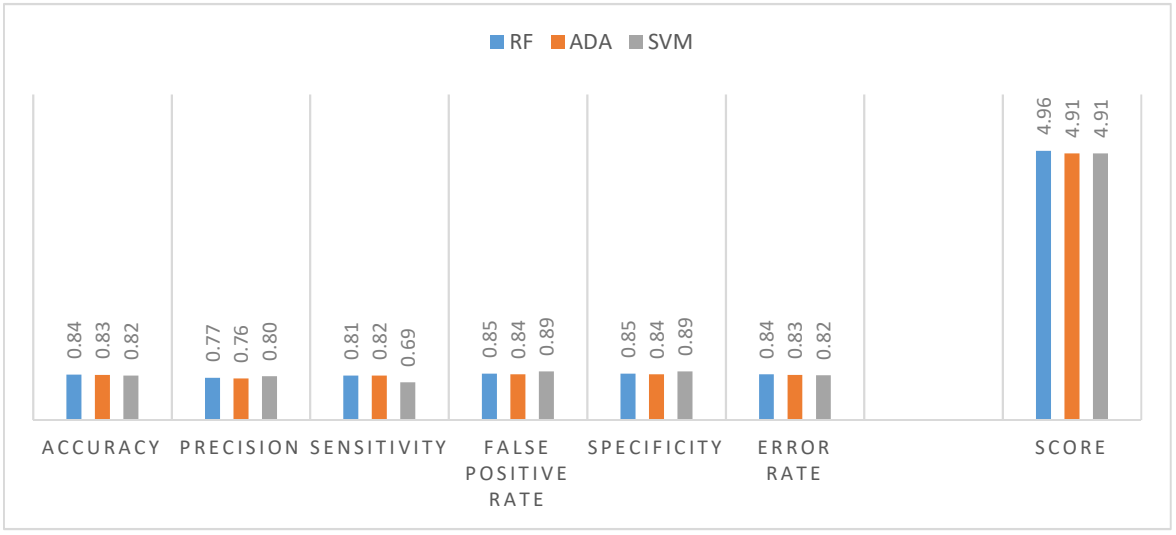


Figure 68: Showing the summary and comparison confusion matrix evaluation results of the RF, ADA and SVM models

The AdaBoost and Support Vector Machine however also scored almost as high and the Random Forest model. The AUC score of the ROC curve for random forest produced 0.914593597161 while AdaBoost and SVM produced 0.908772518228 and 0.89976136873 respectively.

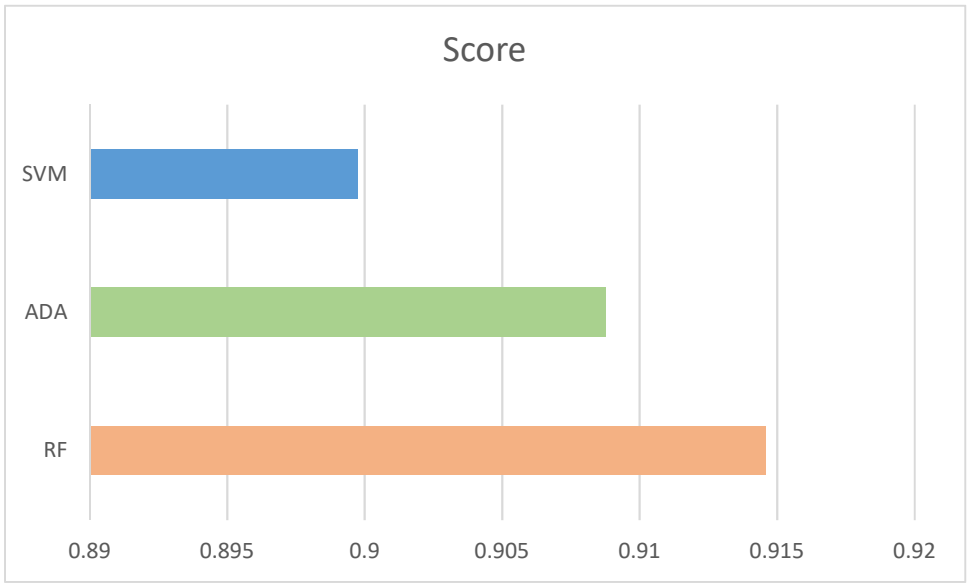


Figure 69: Showing the summary and comparison AUC evaluation results of the RF, ADA and SVM models

The overall results from the evaluation of the three models showed that comparing the three techniques, the RF algorithm outperformed the other two on each evaluation techniques that were used. The other two models generally perform well as their overall scores were closer to the RF performance score. The results of our experiments showed us that the RF model could discern between two assessment classes with a sizable revenue and the rest with some degree of accuracy.

5.2.3 Prototype application development

Using web technologies and resources, we created and implemented a web-based application for tax assessment prediction. The application uses the prediction model we developed using the random forest algorithm using data mining and machine learning. The system is made up of two sub modules which provide two modes of prediction and integration. The primary module is an API web service which provide a connection to more current systems. This module acts as an extension to already existing systems that need to integrate the prediction functionality into their business processes. The API endpoint receives the data in a POST request and un-marshals the input data for further processing. When the data is un-marshalled, it is processed, prepared and passed to the prediction model. The prediction along with the original data is then prepared, encrypted and sent as a response to the client in JSON format. This application can easily be integrated with any system because it uses standard API technology. The prototype API can provide a service to the main tax administration system in Zambia as an extension of the existing audit process of audit case selection. When necessary, users can generate predictions using data using the other module, which is a web-based application with a graphical user interface. The user may choose to make predictions on declaration submitted during a particular period by entering the start and end period on which to run the prediction. The user is presented with a list of taxpayers with their predictions for the specified period. The user also has the option to enter a particular TPIN and run a prediction on it based on the declarations submitted.

5.3 Conclusion

This study's primary goal was to create a prediction model that would aid in the assessment of tax assessments and recommend for audit and assessment. By implementing a system that makes predictions about tax declarations using data mining and machine learning, the goal was to enhance the current tax assessment and audit procedures. The results of the research were able to show the major challenges that affect tax compliance are fraud, taxpayer perceptions,

poor taxpayer education, poor system controls, ignorance and misconceptions among citizens among others. It was found that it is very difficult for tax officers to identify most cases likely to yield significant revenue for the country due to the complex nature of these challenges. In many instances, cases may appear genuine from first glance and using the existing tools. The research looked at how we could use data mining and machine leaning to develop a prediction model to determine audit cases with a potential for significant revenue. The model was developed with a prototype application which demonstrated how the model developed would integrate with the existing system and add value to the current process.

From the information gathered from the revenue authority, it is clear that tax audits play big role in the revenue collection process and a large portion of the revenue collected comes from assessment process in the current systems in place. The potential benefits from improving and making this process more efficient will have a positive impact on the overall revenue collection. The data collected was critically analysed with trends and patterns observed as a preliminary process to the development of prediction model. From the data we were able to make some observations on some patterns appearing in the data. The cases that were marked for assessment feature and the no declaration transactions feature were shown to be strongly correlated. This finding may suggest that taxpayers who fail to submit their declarations on a regular basis are frequently audited and evaluated. This pattern also gives us insight into taxpayer behaviours and beliefs. From the authority's perspective, it could trigger conversations on why taxpayers continue to default in this area from which they may come up with mechanisms which can help alter the general behaviour. This behaviour could indicate that perhaps the assessments and penalties imposed on such kind of defaulters are insignificant deterrent in comparison to overall gain for the taxpayer from their evasion. Such taxpayers may consider it an acceptable loss because of the higher gains achieved from evasion. Another possible explanation to this kind of behaviour could misinformation and lack of information among taxpayers and this becomes a "normal" behaviour. This observation could trigger an increase in taxpayer education and an evaluation of the effectiveness of the current education programs around the country. Others may simply believe the frequency of the audits is acceptable when compared to what they stand to gain from their evasion activities. From the data it was also observed that a high correlation between audits and reassessments for taxpayers who had already undergone assessment. This may indicate that taxpayers who underreport during an audit and assessment are typically repeat offenders. This behaviour also could indicate that the assessments and penalties imposed on such kind of defaulters are insignificant in comparison to overall gain for the taxpayer from

their evasion. Other reasons which may be considered by the revenue authority may be that taxes in the country may be high or considered high in some sectors by the general public thereby leading to repeat offenders getting audits regularly. The authority could investigate such perceptions from the public and if found to have merit may consider realigning the tax schemes in order to redistribute tax burdens among taxpayers. We were also able to observe a relationship between nil values on invoices and assessments in six months and a similar relationship between nil declaration and assessments in the last six months. All this information and tendencies that have been noticed could assist the revenue authority in looking into these trends more thoroughly and implementing strategies and policies to increase taxpayer compliance.

We developed a model and an application with the ability to be integrated into the existing system with an acceptable precision, accuracy and low error rate. According to Barkved [68], it is generally acceptable to have a prediction accuracy of about above 70%. Vidiyala [116] states that a score of 0.5 on the AUC score may be an indication that there was no discrimination form the model. He also suggests that generally a score between 0.5 and 0.7 are considered acceptable results and a score above 0.7 is very good score. The ability of application to integrate with existing systems makes the application helpful as it allows other existing systems with their own internal business processes to leverage on the prediction model and add value without making major changes to their internal systems. One of the challenges faced in the research was the calibre of the information gathered from the source to create the prediction model. This affected accuracy and precision of the model which could improve once the data quality is improved.

5.4 Recommendations

Additional work and research can be done to ride on and complement this research to further improve compliance and taxpayer services. Such additional functions that could be considered include cases such as where a taxpayer failed to turn in their declaration during a specific time frame, the prediction model can be used to detect such cases and an appropriate estimated assessment amount could be derived which would then be posted on the taxpayer's account as a debit entry. In other scenarios, the labour of tax auditors could be reduced by fully automating some of their audit processes and allow the system to post the assessments on the taxpayer's account. One of the most significant issues facing revenue authorities in emerging nations is poor or inadequate data quality. This is normally due to inadequate systems available in the

key strategic institutions of the country. This may include but not limited to software, hardware infrastructure and finances that contribute to better quality data in the country. With an improved overall data quality, more features could be added to the model's training and increase in accuracy and prediction and reduce the error rate. One future development work that could be considered to help overcome this challenge would be to implement automatic reporting from taxpayer POS/CRM systems to the revenue authority. Transactions would be sent to the directly to the revenue authority as soon as the transaction is effected. This process would help the revenue authority get up to date information and help with quicker decision making and improvement in the collection process and raise the standard of the information received from the different taxpayers. Another recommendation to help improve data quality would be to encourage key players in the economy such as government agencies to implement real-time data sharing through interfaces and data exchange agreements. Another recommendation for future work is to build on this research to help the authority gain and use some insights gained from the model to identify areas that require taxpayer education improvement and make targeted programs to help taxpayers.

6. REFERENCES

[1] E. Saez, 'REPORTED INCOMES AND MARGINAL TAX RATES, 1960-2000: EVIDENCE AND POLICY IMPLICATIONS', NATIONAL BUREAU OF ECONOMIC RESEARCH, 2004. Accessed: Feb. 06, 2022. [Online]. Available: https://www.nber.org/system/files/working_papers/w10273/w10273.pdf

[2] R. M. Bird, 'Why tax corporations', Dec. 1996. Accessed: Feb. 22, 2023. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0242274e359f39805844bf32d6dab0efe719748f

[3] E. Auriol and M. Warlters, 'Taxation base in developing countries', *Journal of Public Economics*, vol. 89, no. 4, pp. 625–646, Apr. 2005, doi: 10.1016/j.jpubeco.2004.04.008.

[4] 'Getting to 15 percent: addressing the largest tax gaps'. Accessed: Sep. 15, 2022. [Online]. Available: https://blogs.worldbank.org/governance/getting-15-percent-addressing-largest-tax-gaps

[5] Alfred Mwila, David Manley, Patrick Chileshe, Ezekiel Phiri and Kelvin and Mpembamoto, 'The Taxation System in Zambia', *Final Report*, 2011.

[6] M. R. Palil, 'Tax knowledge and tax compliance determinants in self assessment system in Malaysia', d_ph, University of Birmingham, 2010. Accessed: Sep. 28, 2023. [Online]. Available: https://etheses.bham.ac.uk/id/eprint/1040/

[7] J. E. Anderson, 'Agricultural Use-Value Property Tax Assessment: Estimation and Policy Issues', *Public Budgeting & Finance*, vol. 32, no. 4, pp. 71–94, Dec. 2012, doi: 10.1111/j.1540-5850.2012.01025.x.

[8] Micah Leyira and Chukwuma Ebere, 'Tax System in Nigeria - Challenges and the way forward', Univeristy of Port harcourt.

[9] P. Chander and L. Wilde, 'Corruption in tax administration', *Journal of Public Economics*, vol. 49, no. 3, pp. 333–349, Dec. 1992, doi: 10.1016/0047-2727(92)90072-N.

[10] E. Uslaner, 'Tax Evasion, Corruption, and the Social Contract in Transition', Jan. 2010.

[11] J. A. Torgler Jorge Martinez-Vazquez, Benno, *Developing Alternative Frameworks for Explaining Tax Compliance*. London: Routledge, 2010. doi: 10.4324/9780203851616.

[12] C. Kogler, S. Muehlbacher, and E. Kirchler, 'Testing the "slippery slope framework" among self-employed taxpayers', *Econ Gov*, vol. 16, no. 2, pp. 125–142, May 2015, doi: 10.1007/s10101-015-0158-9.

[13] 'Trends Report - 2021'. Financial Intelligence Center, 2021. [Online]. Available: https://www.fic.gov.zm/79-fic-news/115-trends-report-2021

[14] B. Ariel, 'Deterrence and Moral Persuasion Effects on Corporate Tax Compliance: Findings from a Randomized Controlled Trial*', *Criminology*, vol. 50, no. 1, pp. 27–69, 2012, doi: 10.1111/j.1745-9125.2011.00256.x.

[15] M. G. Allingham and A. Sandmo, 'Income tax evasion: a theoretical analysis', *Journal of Public Economics*, vol. 1, no. 3, pp. 323–338, Nov. 1972, doi: 10.1016/0047-2727(72)90010-2.

[16] W. Nhekairo, 'THE TAXATION SYSTEM IN ZAMBIA', Accessed: Jan. 29, 2022. [Online]. Available: https://www.taxjustice-and-poverty.org/fileadmin/Dateien/Taxjustice_and_Poverty/Zambia/JCTR/JCTR_2014_taxstudy.pdf

[17] C. Kaliba, M. Muya, and K. Mumba, 'Cost escalation and schedule delays in road construction projects in Zambia', *International Journal of Project Management*, vol. 27, no. 5, pp. 522–531, Jul. 2009, doi: 10.1016/j.ijproman.2008.07.003.

[18] W. Easterly, 'How much do distortions affect growth?', *Journal of Monetary Economics*, vol. 32, no. 2, pp. 187–212, Nov. 1993, doi: 10.1016/0304-3932(93)90002-W.

[19] C. R. Blitzer, 'Development and income distribution in a dual economy', *Journal of Development Economics*, vol. 6, no. 3, pp. 407–429, Jan. 1979, doi: 10.1016/0304-3878(79)90024-5.

[20] M. Miskam, M. N. Rohaya, N. Omar, and R. Aziz, 'Determinants of Tax Evasion on Imported Vehicles', *Procedia Economics and Finance*, vol. 7, pp. 205–212, Dec. 2013, doi: 10.1016/S2212-5671(13)00236-0.

[21] Y. Farzami, R. Gregory-Allen, A. Molchanov, and S. Sehrish, 'COVID-19 and the liquidity network', *Finance Research Letters*, vol. 42, p. 101937, Oct. 2021, doi: 10.1016/j.frl.2021.101937.

[22] L. Rakner, 'Tax bargains in unlikely places: The politics of Zambian mining taxes', *The Extractive Industries and Society*, vol. 4, no. 3, pp. 525–538, Jul. 2017, doi: 10.1016/j.exis.2017.04.005.

[23] W. J. Byrne, 'The elasticity of the tax system of Zambia, 1966–1977', *World Development*, vol. 11, no. 2, pp. 153–162, Feb. 1983, doi: 10.1016/0305-750X(83)90065-7.

[24] N. Jayasinghe and M. Ezpeleta, 'Ensuring women follow the money: Gender barriers in extractive industry revenue accountability: The Dominican Republic and Zambia', *The Extractive Industries and Society*, vol. 7, no. 2, pp. 428–434, Apr. 2020, doi: 10.1016/j.exis.2019.12.005.

[25] 'General Tax Information – Zambia Revenue Authority'. Accessed: Jan. 30, 2022. [Online]. Available: https://www.zra.org.zm/tax-information-details/

[26] K. Chanda, 'ZRA Annual report 2020'. Zambia Revenue Authority, Dec. 31, 2020. Accessed: Jan. 30, 2022. [Online]. Available: https://www.zra.org.zm/wp-content/uploads/2021/05/Annual-Report-2020-2.pdf

[27] E. Phiri, 'ZRA Tax Statistics in Zambia 2020'. Zambia Revenue Authority, Dec. 31, 2020. Accessed: Jan. 30, 2022. [Online]. Available: https://www.zra.org.zm/wp-content/uploads/2021/07/Tax-Statistics-2020.pdf

[28] A. O'Sullivan, T. A. Sexton, and S. M. Sheffrin, 'Property Taxes, Mobility, and Home Ownership', *Journal of Urban Economics*, vol. 37, no. 1, pp. 107–129, Jan. 1995, doi: 10.1006/juec.1995.1007.

[29] F. T. Cawood, 'The South African mineral and petroleum resources royalty act—Background and fundamental principles', *Resources Policy*, vol. 35, no. 3, pp. 199–209, Sep. 2010, doi: 10.1016/j.resourpol.2010.03.003.

[30] V. Thuronyi, 'Presumptive Taxation of the Hard-to-Tax', in *Contributions to Economic Analysis*, vol. 268, Elsevier, 2004, pp. 101–120. doi: 10.1016/S0573-8555(04)68805-5.

[31] R. M. Bird and S. Wallace, 'Is it Really so Hard to Tax the Hard-to-Tax? The Context and Role of Presumptive Taxes', in *Contributions to Economic Analysis*, vol. 268, Elsevier, 2004, pp. 121–158. doi: 10.1016/S0573-8555(04)68806-7.

[32] C. Adams and P. Webley, ‘Small business owners’ attitudes on VAT compliance in the UK’, *Journal of Economic Psychology*, vol. 22, no. 2, pp. 195–216, Apr. 2001, doi: 10.1016/S0167-4870(01)00029-0.

[33] M. Keen and B. Lockwood, ‘The value added tax: Its causes and consequences’, *Journal of Development Economics*, vol. 92, no. 2, pp. 138–151, Jul. 2010, doi: 10.1016/j.jdeveco.2009.01.012.

[34] M. Keen and J. Mintz, ‘The optimal threshold for a value-added tax’, *Journal of Public Economics*, vol. 88, no. 3–4, pp. 559–576, Mar. 2004, doi: 10.1016/S0047-2727(02)00165-2.

[35] P. BAKER and V. BRECHLING, ‘The Impact of Excise Duty Changes on Retail Prices in the UK’, *Fiscal Studies*, vol. 13, no. 2, pp. 48–65, 1992.

[36] O. H. Chang, D. R. Nichols, and J. J. Schultz, ‘Taxpayer attitudes toward tax audit risk’, *Journal of Economic Psychology*, vol. 8, no. 3, pp. 299–309, Sep. 1987, doi: 10.1016/0167-4870(87)90025-0.

[37] S. Scotchmer and J. Slemrod, ‘Randomness in tax enforcement’, *Journal of Public Economics*, vol. 38, no. 1, pp. 17–32, Feb. 1989, doi: 10.1016/0047-2727(89)90009-1.

[38] F. Phiri and N. Ndlovu, ‘Tax Compliance - SA Institute of Taxation’. Accessed: May 20, 2022. [Online]. Available: https://www.thesait.org.za/news/524096/Tax-Compliance.htm

[39] S. James and C. Alley, ‘Tax compliance, self-assessment and tax administration’, University Library of Munich, Germany, 26906, 2002. Accessed: Feb. 06, 2022. [Online]. Available: https://ideas.repec.org/p/pra/mprapa/26906.html

[40] T. Besley and T. Persson, ‘Why Do Developing Countries Tax So Little?’, *Journal of Economic Perspectives*, vol. 28, no. 4, pp. 99–120, Nov. 2014, doi: 10.1257/jep.28.4.99.

[41] O. W. Atawodi and S. A. Ojeka, ‘Factors That Affect Tax Compliance among Small and Medium Enterprises (SMEs) in North Central Nigeria’, *IJBM*, vol. 7, no. 12, p. p87, Jun. 2012, doi: 10.5539/ijbm.v7n12p87.

[42] M. Kornhauser, ‘Normative and Cognitive Aspects of Tax Compliance: Literature Review and Recommendations for the Irs Regarding Individual Taxpayers’, *undefined*, 2007, Accessed: Jan. 12, 2022. [Online]. Available: https://www.semanticscholar.org/paper/Normative-and-Cognitive-Aspects-of-Tax-Compliance%3A-Kornhauser/a25be5f64282bfeec186532bfa3bdf757b6ecd53

[43] T. R. Tyler and S. L. Blader, ‘The Group Engagement Model: Procedural Justice, Social Identity, and Cooperative Behavior’, *Pers Soc Psychol Rev*, vol. 7, no. 4, pp. 349–361, Nov. 2003, doi: 10.1207/S15327957PSPR0704_07.

[44] A. Raskolnikov, ‘Crime and Punishment in Taxation: Deceit, Deterrence, and the Self-Adjusting Penalty’, *Columbia Law Review*, vol. 106, no. 3, pp. 569–642, 2006.

[45] I. G. Wallschutzky, ‘Possible causes of tax evasion’, *Journal of Economic Psychology*, vol. 5, no. 4, pp. 371–384, Dec. 1984, doi: 10.1016/0167-4870(84)90034-5.

[46] F. Irawan and A. S. Utama, ‘The Impact of Tax Audit and Corruption Perception on Tax Evasion’, *International Journal of Business and Society*, vol. 22, no. 3, Art. no. 3, Dec. 2021, doi: 10.33736/ijbs.4290.2021.

[47] R. L. Bruno, ‘Tax enforcement, tax compliance and tax morale in transition economies: A theoretical model’, p. 39.

[48] E. Hemberg, J. Rosen, G. Warner, S. Wijesinghe, and U.-M. O'Reilly, 'Tax non-compliance detection using co-evolution of tax evasion risk and audit likelihood', in *Proceedings of the 15th International Conference on Artificial Intelligence and Law*, San Diego California: ACM, Jun. 2015, pp. 79–88. doi: 10.1145/2746090.2746099.

[49] 'Annual report 2022'. Zambia Revenue Authority, Apr. 21, 2023. [Online]. Available: https://www.zra.org.zm/wp-content/uploads/2023/05/ZRA-Annual-Report-2022-compressed.pdf

[50] B. Mahesh, 'Machine Learning Algorithms - A Review', vol. 9, no. 1, p. 7, 2018.

[51] P. Bhavsar, I. Safro, N. Bouaynaya, R. Polikar, and D. Dera, 'Machine Learning in Transportation Data Analytics', in *Data Analytics for Intelligent Transportation Systems*, Elsevier, 2017, pp. 283–307. doi: 10.1016/B978-0-12-809715-1.00012-2.

[52] R. Rastogi and K. Shim, 'PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning', *Data Mining and Knowledge Discovery*, vol. 4, no. 4, pp. 315–344, Oct. 2000, doi: 10.1023/A:1009887311454.

[53] H. Bhavsar and A. Ganatra, 'A Comparative Study of Training Algorithms for Supervised Machine Learning'.

[54] W. Du and Z. Zhan, 'Building Decision Tree Classifier on Private Data', 2002.

[55] 'Everything You Need to Know About Entropy'. Accessed: May 06, 2022. [Online]. Available: https://interestingengineering.com/an-infinite-disorder-the-physics-of-entropy

[56] E. Lisowski, 'What is entropy in machine learning?', Addepto. Accessed: May 06, 2022. [Online]. Available: https://addepto.com/what-is-entropy-in-machine-learning/

[57] N. Tyagi, 'Understanding the Gini Index and Information Gain in Decision Trees', Analytics Steps. Accessed: May 07, 2022. [Online]. Available: https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees-ab4720518ba8

[58] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, 'A comparison of random forest variable selection methods for classification prediction modeling', *Expert Systems with Applications*, vol. 134, pp. 93–101, Nov. 2019, doi: 10.1016/j.eswa.2019.05.028.

[59] M. Belgiu and L. Drăguţ, 'Random forest in remote sensing: A review of applications and future directions', *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, Apr. 2016, doi: 10.1016/j.isprsjprs.2016.01.011.

[60] 'Classification Algorithms - Random Forest'. Accessed: May 10, 2022. [Online]. Available: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_random_forest.htm

[61] G. Biau and E. Scornet, 'A random forest guided tour', *TEST*, vol. 25, no. 2, pp. 197–227, Jun. 2016, doi: 10.1007/s11749-016-0481-7.

[62] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, 'How Many Trees in a Random Forest?', in *Machine Learning and Data Mining in Pattern Recognition*, P. Perner, Ed., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 154–168. doi: 10.1007/978-3-642-31537-4_13.

[63] Y. Qi, 'Random Forest for Bioinformatics', in *Ensemble Machine Learning: Methods and Applications*, C. Zhang and Y. Ma, Eds., Boston, MA: Springer US, 2012, pp. 307–323. doi: 10.1007/978-1-4419-9326-7_11.

[64] L. Breiman, 'Random Forests', *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[65] M. Pal, 'Random forest classifier for remote sensing classification', *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, Jan. 2005, doi: 10.1080/01431160412331269698.

[66] W. S. Noble, 'What is a support vector machine?', *Nat Biotechnol*, vol. 24, no. 12, Art. no. 12, Dec. 2006, doi: 10.1038/nbt1206-1565.

[67] D. A. Pisner and D. M. Schnyer, 'Chapter 6 - Support vector machine', in *Machine Learning*, A. Mechelli and S. Vieira, Eds., Academic Press, 2020, pp. 101–121. doi: 10.1016/B978-0-12-815739-8.00006-7.

[68] E. Tuba and Z. Stanimirovic, 'Elephant herding optimization algorithm for support vector machine parameters tuning', in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Jun. 2017, pp. 1–4. doi: 10.1109/ECAI.2017.8166464.

[69] Y. Zhang *et al.*, 'Research and Application of AdaBoost Algorithm Based on SVM', in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, May 2019, pp. 662–666. doi: 10.1109/ITAIC.2019.8785556.

[70] T.-K. An and M.-H. Kim, 'A New Diverse AdaBoost Classifier', in *2010 International Conference on Artificial Intelligence and Computational Intelligence*, Sanya, China: IEEE, Oct. 2010, pp. 359–363. doi: 10.1109/AICI.2010.82.

[71] L. I. Kuncheva, 'Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy'.

[72] S. Chaudhury, 'Tuning of Adaboost with Computational Complexity', Medium. Accessed: Nov. 07, 2023. [Online]. Available: https://medium.com/@chaudhurysrijani/tuning-of-adaboost-with-computational-complexity-8727d01a9d20

[73] C. Milner and B. Berg, 'Tax Analytics Artificial Intelligence and Machine Learning–Level 5'. Accessed: Jan. 23, 2022. [Online]. Available: https://www.pwc.no/no/publikasjoner/Digitalisering/artificial-intelligence-and-machine-learning-final1.pdf

[74] H. Höglund, 'Tax payment default prediction using genetic algorithm-based variable selection', *Expert Systems with Applications*, vol. 88, pp. 368–375, Dec. 2017, doi: 10.1016/j.eswa.2017.07.027.

[75] '(PDF) Tax declaration rates via audits: a prediction using Markov model'. Accessed: Feb. 09, 2024. [Online]. Available: https://www.researchgate.net/publication/319908643_Tax_declaration_rates_via_audits_a_prediction_using_Markov_model

[76] M. Rabasco and P. Battiston, 'Predicting the deterrence effect of tax audits. A machine learning approach', *Metroeconomica*, vol. 74, Mar. 2023, doi: 10.1111/meca.12420.

[77] V. Baghdasaryan, H. Davtyan, A. Sarikyan, and Z. Navasardyan, 'Applied Artificial Intelligence Improving Tax Audit Efficiency Using Machine Learning: The Role of Taxpayer's Network Data in Fraud Detection', *Applied Artificial Intelligence*, vol. 36, Jan. 2022, doi: 10.1080/08839514.2021.2012002.

[78] Z. Cui, 'China's Export Tax Rebate Policy', *China: An International Journal*, vol. 1, no. 2, pp. 339–349, 2003, doi: 10.1353/chn.2005.0035.

[79] M. Andini, E. Ciani, G. de Blasio, A. D'Ignazio, and V. Salvestrini, 'Targeting with machine learning: An application to a tax rebate program in Italy', *Journal of Economic Behavior & Organization*, vol. 156, pp. 86–102, Dec. 2018, doi: 10.1016/j.jebo.2018.09.010.

[80] H. S. Seippel and M. Thesis, 'Customer purchase prediction through machine learning', p. 95.

[81] L. Ying, 'Research on bank credit default prediction based on data mining algorithm', *The International Journal of Social Sciences and Humanities Invention*, vol. 5, pp. 4820–4823, Jun. 2018, doi: 10.18535/ijsshi/v5i6.09.

[82] P. Pompe and J. Bilderbeek, 'The Prediction of Bankruptcy of Small-and-Medium Sized Industrial Firms', *Journal of Business Venturing*, vol. 20, pp. 847–868, Nov. 2005, doi: 10.1016/j.jbusvent.2004.07.003.

[83] S. F., 'Machine-Learning Techniques for Customer Retention: A Comparative Study', *ijacsa*, vol. 9, no. 2, 2018, doi: 10.14569/IJACSA.2018.090238.

[84] S. Yilmazer and S. Kocaman, 'A mass appraisal assessment study using machine learning based on multiple regression and random forest', *Land Use Policy*, vol. 99, p. 104889, Dec. 2020, doi: 10.1016/j.landusepol.2020.104889.

[85] J. Lismont *et al.*, 'Predicting tax avoidance by means of social network analytics', *Decision Support Systems*, vol. 108, pp. 13–24, Apr. 2018, doi: 10.1016/j.dss.2018.02.001.

[86] C. Schröer, F. Kruse, and J. M. Gómez, 'A Systematic Literature Review on Applying CRISP-DM Process Model', *Procedia Computer Science*, vol. 181, pp. 526–534, Jan. 2021, doi: 10.1016/j.procs.2021.01.199.

[87] S. Moro, R. M. S. Laureano, and P. Cortez, 'USING DATA MINING FOR BANK DIRECT MARKETING: AN APPLICATION OF THE CRISP-DM METHODOLOGY', p. 5.

[88] R. Wirth and J. Hipp, 'CRISP-DM: Towards a Standard Process Model for Data Mining', p. 11.

[89] J. A. Solano, D. J. Lancheros Cuesta, S. F. Umaña Ibáñez, and J. R. Coronado-Hernández, 'Predictive models assessment based on CRISP-DM methodology for students performance in Colombia - Saber 11 Test', *Procedia Computer Science*, vol. 198, pp. 512–517, Jan. 2022, doi: 10.1016/j.procs.2021.12.278.

[90] Kirsten Barkved, 'How To Know if Your Machine Learning Model Has Good Performance | Obviously AI'. Accessed: Nov. 01, 2023. [Online]. Available: https://www.obviously.ai/post/machine-learning-model-performance

[91] J. Spacey, '4 Examples of System Architecture', Simplicable. Accessed: May 21, 2022. [Online]. Available: https://simplicable.com/new/system-architecture

[92] ‘Python Tutorial’, tutorialspoint. Accessed: May 12, 2022. [Online]. Available: https://www.tutorialspoint.com/python/index.htm

[93] M. Lowry, A. Philpot, T. Pressburger, and I. Underwood, ‘Amphion: Automatic programming for scientific subroutine libraries’, in *Methodologies for Intelligent Systems*, Z. W. Raś and M. Zemankova, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1994, pp. 326–335. doi: 10.1007/3-540-58495-1_33.

[94] J. Hao and T. K. Ho, ‘Machine Learning Made Easy: A Review of *Scikit-learn* Package in Python Programming Language’, *Journal of Educational and Behavioral Statistics*, vol. 44, no. 3, pp. 348–361, Jun. 2019, doi: 10.3102/1076998619832248.

[95] T. G. Dietterich, ‘Ensemble Methods in Machine Learning’, in *Multiple Classifier Systems*, vol. 1857, in Lecture Notes in Computer Science, vol. 1857. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15. doi: 10.1007/3-540-45014-9_1.

[96] W. McKinney, ‘pandas: powerful Python data analysis toolkit’, *08/11/2012*.

[97] J. D. Hunter, ‘Matplotlib: A 2D Graphics Environment’, *Computing in Science & Engineering*, vol. 9, no. 03, pp. 90–95, May 2007, doi: 10.1109/MCSE.2007.55.

[98] E. Bisong, ‘Matplotlib and Seaborn’, in *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, E. Bisong, Ed., Berkeley, CA: Apress, 2019, pp. 151–165. doi: 10.1007/978-1-4842-4470-8_12.

[99] M. Waskom, ‘seaborn: statistical data visualization’. Jun. 04, 2021.

[100] ‘Understanding Python Pickling with example’, GeeksforGeeks. Accessed: May 13, 2022. [Online]. Available: https://www.geeksforgeeks.org/understanding-python-pickling-example/

[101] F. Aslam and H. Mohammed, ‘Efficient Way Of Web Development Using Python And Flask’, *ijarcs*, 2015.

[102] S. Suraya and M. Sholeh, ‘Designing and Implementing a Database for Thesis Data Management by Using the Python Flask Framework’, *International Journal of Engineering, Science and Information Technology*, vol. 2, no. 1, Art. no. 1, 2022, doi: 10.52088/ijesty.v2i1.197.

[103] ‘Flask-RESTful — Flask-RESTful 0.3.8 documentation’. Accessed: May 15, 2022. [Online]. Available: https://flask-restful.readthedocs.io/en/latest/

[104] ‘What is REST?’, Codecademy. Accessed: May 15, 2022. [Online]. Available: https://www.codecademy.com/article/what-is-rest

[105] J. F. Pimentel, L. Murta, V. Braganholo, and J. Freire, ‘A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks’, in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, May 2019, pp. 507–517. doi: 10.1109/MSR.2019.00077.

[106] ‘Visual Studio Code Frequently Asked Questions’. Accessed: May 15, 2022. [Online]. Available: https://code.visualstudio.com/docs/supporting/faq

[107] H. Du, P. Jones, E. L. Segarra, and C. F. Bandera, ‘Development of a REST API for obtaining site-specific historical and near-future weather data in EPW format’, presented at the Building Simulation and Optimization 2018, Emmanuel College, University of Cambridge, Sep. 2018. Accessed: May 15, 2022. [Online]. Available: https://orca.cardiff.ac.uk/id/eprint/111475/

[108] ‘Anaconda Navigator — Anaconda documentation’. Accessed: May 15, 2022. [Online]. Available: https://docs.anaconda.com/anaconda/navigator/index.html

[109] ‘Random Forest Parameter Tuning | Tuning Random Forest’, Analytics Vidhya. Accessed: May 17, 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/

[110] J. R. Kreiger, ‘Evaluating a Random Forest model’, Analytics Vidhya. Accessed: May 17, 2022. [Online]. Available: https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56

[111] Y. Li, N. R. Katsipoulakis, B. Chandramouli, J. Goldstein, and D. Kossmann, ‘Mison: a fast JSON parser for data analytics’, *Proc. VLDB Endow.*, vol. 10, no. 10, pp. 1118–1129, Jun. 2017, doi: 10.14778/3115404.3115416.

[112] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, ‘Foundations of JSON Schema’, in *Proceedings of the 25th International Conference on World Wide Web*, in WWW ’16. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2016, pp. 263–273. doi: 10.1145/2872427.2883029.

[113] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, ‘A concrete security treatment of symmetric encryption’, in *Proceedings 38th Annual Symposium on Foundations of Computer Science*, Oct. 1997, pp. 394–403. doi: 10.1109/SFCS.1997.646128.

[114] A. Joy, ‘A Deep Dive Into Fernet Module in Python’, Pythonista Planet. Accessed: May 18, 2023. [Online]. Available: https://pythonistaplanet.com/fernet/

[115] R. Vidiyala, ‘Performance Metrics for Classification Machine Learning Problems’, Medium. Accessed: Nov. 02, 2023. [Online]. Available: https://towardsdatascience.com/performance-metrics-for-classification-machine-learning-problems-97e7e774a007

[116] R. Vidiyala, ‘Performance Metrics for Classification Machine Learning Problems’, Medium. Accessed: Nov. 01, 2023. [Online]. Available: https://towardsdatascience.com/performance-metrics-for-classification-machine-learning-problems-97e7e774a007

7. PUBLICATIONS

[1] W.A. Sampa, J. Phiri, ‘Prediction Model for Tax Assessments Using Data Mining and Machine Learning’ Artificial Intelligence Application in Networks and Systems, 2023 | Book chapter | Author DOI: 10.1007/978-3-031-35314-7_1