RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS

By

Chew Chun Phang

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS)

DIGITAL ECONOMY TECHNOLOGY

Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS

Academic Session: JUNE 2024

I CHEW CHUN PHANG

(CAPITAL LETTER)

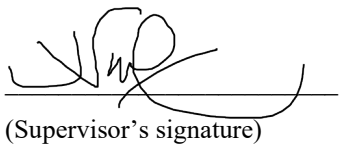
declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)



(Supervisor's signature)

Address:

6, Persiaran Halaman Ampang 24,

Halaman Ampang Mewah,

31400 Ipoh, Perak

NURUL SYAFIDAH JAMIL

Supervisor's name

Date: 3/9/2024

Date: 11/9/2024

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

UNIVERSITI TUNKU ABDUL RAHMAN FACULTY OF

Information and Communication Technology

Date: 3/9/2024_____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that _____Chew Chun Phang_____ (ID No: 2200944) has completed this final year project/ dissertation/ thesis* entitle"**RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS**" under the supervision of Cik Nurul Syafidah Binti Jamil (Supervisor) from the Department of Digital Economy Technology, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

_____

(Chew Chun Phang)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : CHEW CHUN PHANG

Date : 3/9/2024

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Miss Nurul Syafidah binti Jamil who has given me this bright opportunity to engage in "RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS". As someone with no prior background in these subjects, I am grateful for the guidance and support provided by him in navigating this complex field.

I would also like to extend my gratitude to my family and friends for their unwavering support throughout this project. In particular, I would like to thank my parents for their constant encouragement and belief in my abilities.

ABSTRACT

In changing dynamic financial risk management, this project seeks to advance credit scoring predictions through the sentiment analysis into the data science framework, with a specific focus on Natural Language Processing (NLP) and classification algorithms. Traditional credit scoring models, they rely on the traditional historical financial data, normally cannot capture the real-time dynamics and external factors that can impact the borrower's creditworthiness. The objective is to use the power of sentiment analysis, originate from the diverse textual sources such as social media and financial reports to increase the accuracy and flexibility of credit risk assessments. The project adopts a development-based approach with field of data science, leveraging NLP techniques and classification algorithms to seamlessly integrate sentiment-derived features with conventional credit scoring attributes. The methodology emphasizes the fusion of sentiment-derived insights with established credit data, ensuring a comprehensive understanding of credit risk factors. The methodology involves five steps to process data; those are data collection, text preparation, sentiment detection, sentiment classification, and presentation of output. This is to ensure the accuracy of the borrower's creditworthiness will increase compared to the traditional credit scoring models. This proposal will discuss a few relevant topics such as literature reviews, research analysis, and conclusion of the project work.

Table of Contents

TITLE PAGE i

REPORT STATUS DECLARATION FORM ii

FYP THESIS SUBMISSION FORM iii

DECLARATION OF ORIGINALITY iv

ACKNOWLEDGEMENTS v

ABSTRACT vi

TABLE OF CONTENTS vii

LIST OF FIGURES x

LIST OF TABLES xi

LIST OF ABBREVIATIONS xiii

CHAPTER 1 INTRODUCTION **1**

Problem Statement and Motivation 3

1.2 Project Scope 4

1.3 Project Objectives 5

1.4 Contributions 6

1.5 Report Organization 7

CHAPTER 2 LITERATURE REVIEW **8**

2.1Review of the Technologies 8

2.2 Literature Review 10
2.2.1 CalcXML: Credit Scoring Calculator 10
2.2.2 AMEX: Credit Score Model 12
2.2.3 Credit Scoring Methods: Latest Trends and Points to Consider 14

CHAPTER 3 SYSTEM METHODOLOGY/APPROACH **16**

3.0 Methodology 16

3.1 System Design Diagram/Equation 18
3.1.1 System Architecture Diagram 18

3.1.2 Use Case Diagram and Description 20
3.1.3 Activity Diagram 21

CHAPTER 4: SYSTEM DESIGN 22

4.1 Sentiment Analysis 22

4.2 Modelling Block Diagram 24

4.3 Modelling flow 25

4.4 Weight of credit score and polarity 27

4.5 Credit band 27

4.6 Type of classifier 28

CHAPTER 5: SYSTEM IMPLEMENTATION 33

5.1 Hardware Setup 33

5.2 Software Setup 33

5.3 Setting and Configuration 35

5.4 Data Preprocessing 36
5.4.1 Data Transformation 40
5.4.2 Data Visualization 43

5.5 FINBERT 46

5.6 Model Training 47

5.7 Model Evaluation 49

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION 51

6.1 Interface Design 51

6.2 Maths 53

6.3 System Testing and Performance Metrics 54

6.4 Testing Setup and Result 54

6.4 Project Challenges 56

6.5 Objectives Evaluation 56

CHAPTER 7 CONCLUSION AND RECOMMENDATION 57

7.1 Conclusion 57

7.2 Recommendations 57

REFERENCES 58

WEEKLY REPORT 59

PLAGIARISM CHECK RESULT 64

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1.1	Alternative Credit Scoring Comparison	1
Figure 2.1.1	CalcXML	5
Figure 2.1.2	AMEX: Credit Score Model.	7
Figure 2.1.3	Credit Scoring Methods: Latest Trends and Points to Consider	9
Figure 3.0	Agile Methodology.	14
Figure 3.3.1	System Design	18
Figure 3.1.2.1	Use Case Diagram	20
Figure 3.1.3	Activity Diagram	21
Figure 3.3.1.1	Sentiment Analysis	22
Figure 4.2.1	Modelling	24
Figure 4.6.1	Decision Tree	28
Figure 4.6.2	Random Forest	29
Figure 4.6.3	KNN	30
Figure 4.6.4	Gaussian Naive Bayes	31
Figure 4.6.5	XGBoost	32
Figure 5.2.1	Jupyter Notebook	33
Figure 5.2.2	VS Code	34
Figure 5.2.3	Flask	34
Figure 5.4.1	Function to Perform Cleaning	36
Figure 5.4.2	Label Encoding, Normalize Data	37
Figure 5.4.3	Text Cleaning	38

Figure 5.4.4	Text Preprocessing	38
Figure 5.4.5	Using Vectorizer	39
Figure 5.4.6	Lemmatized Text	39
Figure 5.4.1.1	Data Transform	40
Figure 5.4.1.2	Drop Columns	41
Figure 5.4.1.3	LabelEncoder	41
Figure 5.4.1.4	Check for Null	42
Figure 5.4.1.5	Credit Score	42
Figure 5.4.2.1	Month and Credit Score Distribution	43
Figure 5.4.2.2	Occupation and Credit Score Distribution	43
Figure 5.4.2.3	Credit Mix and Credit Score Distribution	44
Figure 5.4.2.4	Income Distribution	45
Figure 5.5.1	FINBERT	46
Figure 5.6.1	Import Library	47
Figure 5.6.2	Function to Evaluate Model	47
Figure 5.6.3	Define Parameter Distribution	48
Figure 5.6.4	Model Comparison	48
Figure 5.7.1	Best Model	49
Figure 5.7.2	Feature Importance	50
Figure 6.1.1	Interface	51
Figure 6.1.3	Interface Logic	52
Figure 6.2.1	Calculation of Credit Score	53
Figure 6.4.1	Sample Data 1	54
Figure 6.4.2	Sample Data 2	55

LIST OF TABLES

Table 5.1	Specifications of laptop	33
Table 6.4.1	result of sample data 1	55
Table 6.4.2	result of sample data 2	55

LIST OF ABBREVIATIONS

NLP	Natural language processing
SVM	Support Vector Machine
ML	Machine Learning
Nan	Not-a-Number

CHAPTER 1 Introduction

Introduction

With advancements in technology nowadays, there are many companies want to adopt sentiment analysis to understand their customers' sentiments [1]. Sentiment analysis involves examining the positive or negative expressions within textual content to gauge customer sentiment. Through contextual analysis, businesses can gain insights into the social sentiments of their customers by actively monitoring online conversations [2]. The traditional credit scoring is relied on the common model that based on the financial data, credit report, standardized metrics, and the problem is may not be suitable for everyone. Ginimachine website reported that America have over 28 million people are credit invisible and 21 million who are considered as not scoreable under the traditional credit scoring calculation [3]. The statistics already show that traditional credit scoring it's doesn't work for everyone. The alternative credit scoring is the method that enhanced from traditional credit scoring models, it's included like utility, rent payment, mobile payment, social media activity and other behavioral pattern.

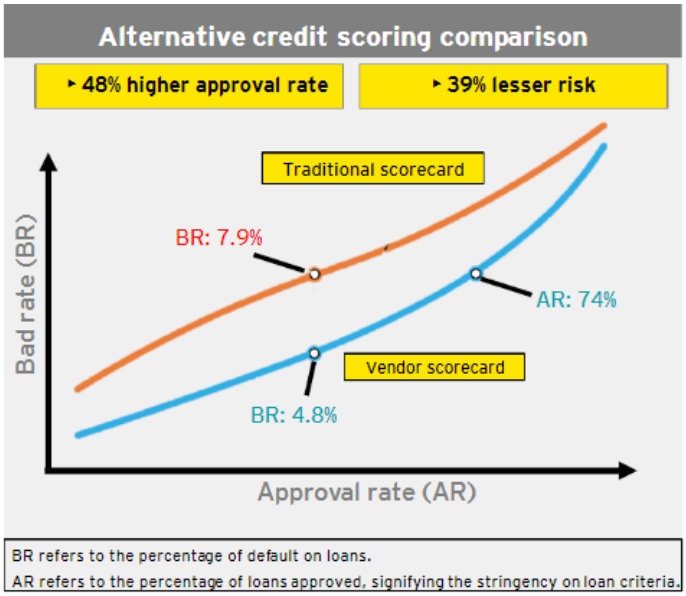


Figure 1.1.1 Alternative Credit Scoring Comparison [4]

Besides that, the reasons that alternative credit scoring is important due to alternative credit data enables lenders to broaden their services to individuals deemed "credit-invisible" – those who were previously ineligible for loans under the conventional credit scoring system. Traditional credit data reports heavily emphasize an individual's credit history as a decisive factor in scoring. Consequently, individuals lacking a credit history face significant challenges in qualifying for new credit. The introduction of alternative data enhances the prospects for credit-invisible consumers to secure a loan, offering them improved chances of accessing credit facilities [5].

In the pursuit of financial inclusivity, particularly among the underbanked and unbanked populations residing in remote areas with limited access to traditional banking services, the incorporation of sentiment analysis emerges as a game-changing mechanism. Many individuals in such regions encounter challenges in opening conventional bank accounts due to geographical constraints and a lack of infrastructure. However, the integration of sentiment analysis into alternative credit scoring models, applied through online platforms, signifies a significant step towards enhanced financial accessibility. By scrutinizing behavioral patterns gleaned from social media platforms like Twitter, sentiment analysis provides nuanced insights into individuals' financial behaviors, surpassing the limitations of traditional credit scoring metrics [6]. This innovative approach not only facilitates credit access for those historically excluded from mainstream financial services but also contributes to cultivating a more inclusive financial landscape that transcends geographical constraints.

Problem Statement and Motivation

Problem Statement

Underrepresentation of Credit-Invisible Individuals:

The conventional credit scoring models predominantly rely on historical credit data, making it challenging for individuals without a credit history, often referred to as "credit-invisible," to qualify for loans. This problem excludes a significant demographic from accessing credit facilities. The project aims to address the underrepresentation of credit-invisible individuals by incorporating sentiment analysis, providing a more inclusive credit scoring methodology that considers alternative data sources beyond traditional credit histories.

Transparency and Interpretability in Credit Scoring Decisions:

Traditional credit scoring models often lack transparency, leading to concerns about the interpretability of the factors influencing credit risk predictions. This poses a challenge for stakeholders, including financial institutions and borrowers, who seek a clearer understanding of the decision-making process. The project addresses this problem by focusing on developing a credit scoring model that not only enhances predictive accuracy but also introduces transparency and interpretability into the risk management process.

Real-time Adaptability in Credit Scoring Models:

The existing credit scoring models employed in risk management often lack real-time adaptability, relying heavily on historical financial data. This limitation poses a challenge in promptly responding to dynamic economic conditions and external influences. The project aims to address the problem of delayed risk assessments by incorporating sentiment analysis, ensuring a more responsive and adaptive credit scoring model.

Motivation

The motivation behind our project, "Risk Management Credit Scoring Prediction Using Sentiment Analysis," is rooted in a desire to address problems with traditional credit scoring. Many people, particularly those without a typical credit history (known as "credit-invisible"), face challenges in qualifying for loans due to the heavy reliance on past financial data. This exclusionary practice motivated this project to explore a more inclusive credit scoring method by incorporating sentiment analysis. Additionally, traditional credit scoring models are often seen as complex and unclear, creating uncertainty for both lenders and borrowers. The project is driven by the goal of making credit evaluations more transparent and understandable by developing a credit scoring model that not only improves accuracy but also adapts in real-time to changes in the economy. This project aims to create a system that considers a wider range of data sources, including sentiments expressed online, to empower individuals and make the credit evaluation process fairer and more accessible.

1.2 Project Scope

The objective of the project is to develop a credit scoring model with sentiment analysis for financial sector to increase the financial inclusion and develop a dashboard for the users to monitor their financial status.

The Key features of the project include:

- The credit scoring model will be able to show the percentage of the users' credit scoring real time.
- The credit scoring model will be able calculate the user credit score by using polarity based on the context of the social media.
- The credit scoring model show the user overall majority behavior patterns by analyzing the context of the social media.
- The credit scoring model will provide credit band based on the credit score.

1.3 Project Objectives

The main goal of this project is to create a smarter way to assess credit risk by combining traditional credit scoring with sentiment analysis. The aim is to improve how we predict creditworthiness by considering people's sentiments. This approach seeks to make the credit scoring process more accurate, adaptable, and transparent. Ultimately, the project aims to contribute to a stable and fair financial system by addressing the limitations of current credit scoring models. The objective aim to have:

Determine the Polarity of User Context or Comment:

The primary objective is to assess and understand the sentiment expressed in user comments or contexts. This involves identifying whether the language used is positive, negative, or neutral. Implement Natural Language Processing (NLP) techniques to analyze textual data. Utilize sentiment analysis algorithms to evaluate the sentiment expressed by users in comments, feedback, or any textual input. A sentiment analysis model capable of categorizing user context into positive, negative, or neutral sentiments.

Develop a System to Calculate Credit Score Based on Polarity:

Create a robust algorithm that calculates the overall polarity of a user based on various inputs. This involves considering multiple factors contributing to the user's sentiment and deriving a comprehensive polarity score. The polarity score will be use in the calculation of the credit scoring. The credit limit will be adjusted based on the credit score.

Develop a Web for Real-Time Prediction:

Build a user-friendly web that provides real-time prediction and visualization of sentiment-related metrics. This dashboard should offer insights into the overall majority sentiment counts, credit band, credit score and overall the sentiment of the texts.

1.4 Contributions

The contributions of the project, "Risk Management Credit Scoring Prediction Using Sentiment Analysis," extend far beyond traditional credit scoring methodologies. By addressing the underrepresentation of credit-invisible individuals and introducing sentiment analysis, the project significantly contributes to increasing financial inclusion. This means that individuals who were previously excluded due to a lack of conventional credit history now have enhanced opportunities to qualify for loans and participate in financial activities. Additionally, the model places a strong focus on transparency and interpretability in credit scoring determinations, thereby empowering users. This is achieved through the incorporation of alternative data sources, specifically social media data. It allows individuals to easily monitor and understand their financial status, fostering financial literacy and providing a clearer view of the factors influencing credit risk predictions. Overall, these contributions mark a transformative step in the financial sector, promoting fairness, accessibility, and empowerment for a more inclusive and informed financial landscape.

1.5 Report Organization

In Chapter 1, the project introduction is provided the overview of the project including the problem statement and motivation, scope, to point out the issues that will affect my project. We also point out the objectives of the project to have a clear view what are the aims of the project that need to be achieve and project scope is to lead us to the correct direction. Lastly, it also highlighted the contribution to the individuals and society and provide a report organization to provide a roadmap for the reader what we going to carry out in this project.

Chapter 2, the literature reviews which is the existing program or research related to the project and we need to understand the way of the current system do and comparison between the selected program and research.

Chapter 3, which is talking about the system methodology that needs to be for development-based project. It also mentions about the system design diagram, use case diagram, and activity diagram to providing a crystal-clear visualization for reader to understand the whole project structure and flow. It also lists out the implementation issues and challenges and timeline to let readers know the estimated timeline for deliverable and milestones of the project.

Chapter 4, the preliminary work done and show the results according to the plan that we proposed in previous chapter, and we will highlight the feasibility of the method we proposed.

Chapter 5, the conclusion, is the summary of the project current process including the problem, motivation, and proposed solution.

CHAPTER 2 LITERATURE REVIEW

2.1Review of the Technologies

2.1.1 Hardware

For my project, I'm working on an HP Omen Series laptop, which is equipped with an Intel Core i7-7700HQ processor. The laptop runs on Windows 10, providing a stable environment for all tasks. It has an NVIDIA GeForce GTX 1050 graphics card, which handles graphics-intensive applications smoothly. The system also includes 8GB of DDR4 RAM, allowing for efficient multitasking, and a 1TB PCIE SSD, which offers plenty of storage space and fast data access, making it a reliable setup for all my computing needs.

2.1.2 Firmware/OS

In our project, we've set up a development environment on Windows 10, which provides a reliable platform for all our work. For coding, we use VS Code and Jupyter Lab. VS Code is great for writing and testing code, while Jupyter Lab is particularly useful for running Python scripts and working with data. To bring everything together into a web application, we rely on Flask, a simple yet effective Python framework that allows us to create and manage web-based interfaces for our model. This setup supports the smooth development and deployment of our credit scoring prediction model.

2.1.3 Programming Language

The project will adpot Python as the programming language because it is more user-friendly and has a wide range of libraries that cater to diverse needs. We also use Flask, which is a web framework written in Python, as it helps in building and managing the web application part of the project effectively. Thus, using Python in conjunction with Flask provides enough power and freedom needed to implement the solutions. With the availability of open-source libraries and software packages for sentiment analysis, such as NLTK, scikit-learn, and TensorFlow

2.1.4 Algorithms

The project will use FinBERT algorithm, a specialized version of the BERT model that trained and nice fit for financial applications or sector. FinBERT is particularly useful for tasks such as sentiment analysis within financial texts, so it's a very important component to our credit scoring prediction model. By leveraging FinBERT, we can analyze and interpret financial language more accurately, enhancing the overall performance and reliability of our model.

2.1.5 Summary of the Technologies Review

In summary, the credit scoring prediction is built on the solid hardware and software foundation which can ensure the system can operate as smooth as possible. The project relies on python and flask python to build a user-friendly web application. The main element of credit scoring prediction is based on the traditional data and sentiment analysis based on the Finbert algorithm, ensuring precise sentiment analysis and improved model performance.

2.2 Literature Review

Review on Risk Management Credit Scoring

2.2.1 CalcXML: Credit Scoring Calculator

The CalcXML provides a valuable tool for individuals seeking to estimate their credit scores. This user-friendly credit score calculator offers a straightforward and accessible way for users to gauge their creditworthiness based on essential financial information. Users input data such as their outstanding balances, credit limits, and payment history, and the calculator generates an estimate of their credit score. This resource serves as a practical and informative aid for those looking to understand how certain financial behaviors may impact their credit standing. The calculator's simplicity and clarity make it a useful platform for individuals keen on monitoring and improving their credit scores, contributing to financial literacy and informed financial decision-making [7].

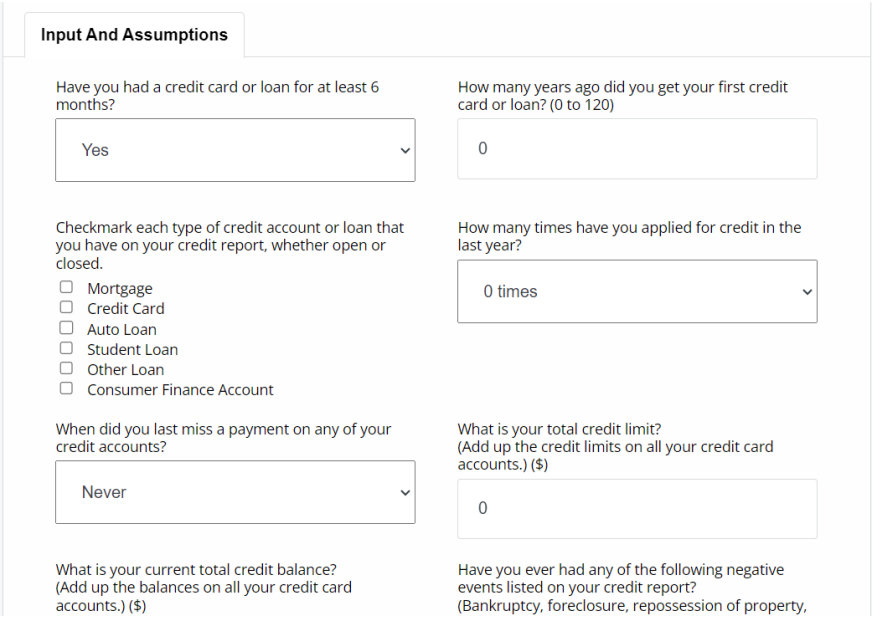


Figure 2.1.1 CalcXML

Strengths:

User-Friendly Interface: The website boasts a user-friendly interface that simplifies the credit score calculation process. Users can easily input their financial details into the calculator, making it accessible to a wide range of individuals.

Educational Value: The credit score calculator provides educational value by helping users understand the relationship between their financial behaviors and credit scores. It serves as an informative tool for those looking to enhance their financial literacy.

Quick Estimation: The calculator offers a swift estimation of credit scores based on the entered information. This speed is advantageous for users seeking a rapid assessment of their credit standing.

Weaknesses:

Simplicity Limits Accuracy: While the calculator is straightforward, its simplicity may limit the accuracy of credit score estimations. It may not capture the full complexity of credit scoring algorithms used by credit bureaus, providing only a basic approximation.

Limited Data Fields: The calculator relies on a limited set of financial data fields for its estimations. This simplicity might overlook some of the nuanced factors that traditional credit scoring models consider, potentially leading to less comprehensive results.

Lack of Customization: The calculator may lack customization options, such as incorporating specific details unique to individual financial situations. This limitation could impact the precision of the credit score estimate, as it may not account for diverse financial scenarios.

Absence of Sentiment Analysis: One drawback of the credit score calculator is that it doesn't consider sentiment analysis. Sentiment analysis, which looks at online behaviors and social media, gives a more complete picture of someone's financial habits. Without this feature, the calculator might miss important factors that influence credit scores, making its estimations less detailed and accurate. Adding sentiment analysis could help capture a broader view of someone's financial behavior and improve the calculator's precision in evaluating creditworthiness.

2.2.2 AMEX: Credit Score Model

The Kaggle project titled "Amex Credit Score Model" provides an intriguing exploration into the development of a credit score model, leveraging the dataset and resources available on the Kaggle platform. The project appears to focus on creating a predictive model for credit scores, a crucial component in the financial sector. By analysing the dataset, the project likely delves into feature engineering, model training, and evaluation techniques to enhance the accuracy of credit score predictions. Kaggle, being a collaborative platform for data science enthusiasts, facilitates knowledge sharing and allows for the exchange of insights. While the specific details of the model's methodology and findings are not provided in the link, the project's premise aligns with the broader goal of utilizing data-driven approaches to optimize credit scoring systems, potentially contributing to advancements in risk management within the financial industry [8].

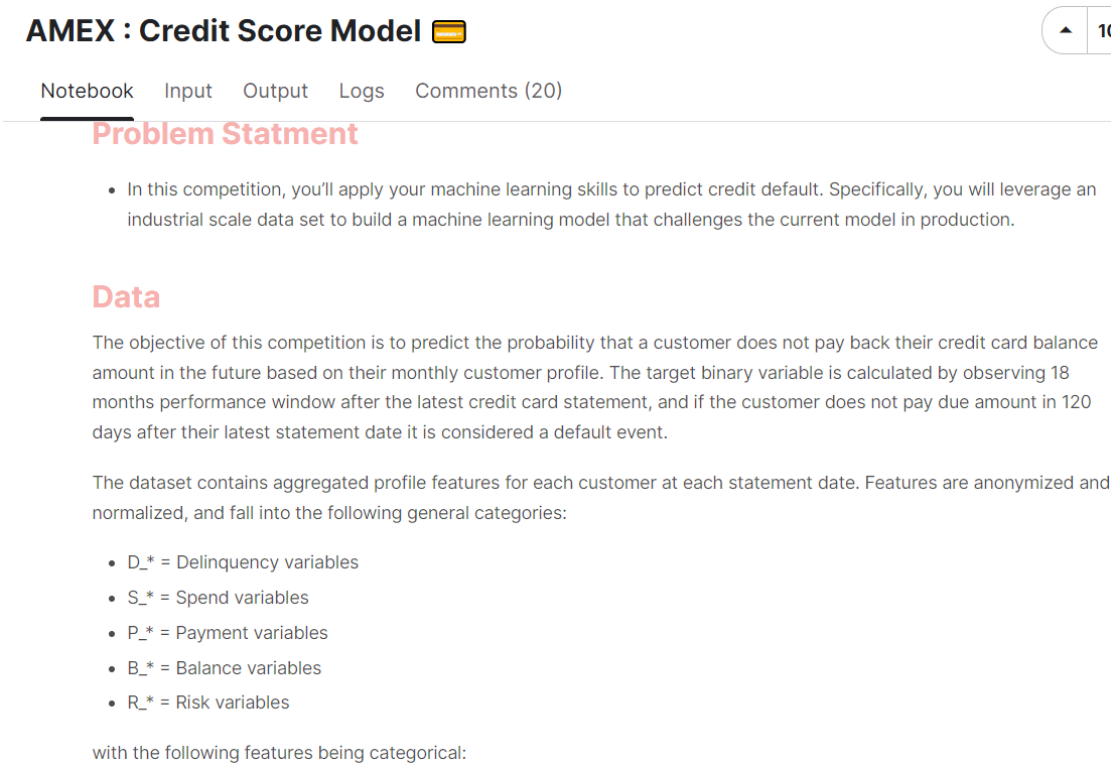


Figure 2.1.2 AMEX: Credit Score Model

Strengths:

Comprehensive Dataset Utilization: One notable strength of the "Amex Credit Score Model" project is the comprehensive utilization of the dataset available on Kaggle. Effectively leveraging the dataset allows for a more robust credit score model, potentially leading to enhanced predictive accuracy. This strength underscores the project's commitment to utilizing rich data sources to inform and improve the credit scoring process.

Feature Engineering Expertise: The "Amex Credit Score Model" project showcases strength in feature engineering, a crucial aspect of building robust predictive models. Effective feature engineering involves selecting, transforming, or creating relevant features from the dataset, contributing significantly to the model's ability to capture patterns and make accurate credit score predictions.

Weaknesses:

Limited Explanation of Model Choices: One weakness is the lack of detailed explanations regarding the specific choices made in the model-building process. Without insights into why certain algorithms or parameters were chosen, users may find it challenging to comprehend the rationale behind the model's architecture. Improved documentation on the decision-making process could enhance the project's educational value and assist other users in understanding and replicating the approach.

Potential Overfitting Concerns: The project might face a weakness related to potential overfitting, especially if the model is highly tuned to the training data. Overfitting occurs when a model performs well on the training set but struggles to generalize to new, unseen data. Clear indications of steps taken to address overfitting concerns or enhance model generalization would strengthen the project's reliability and applicability beyond the provided dataset.

2.2.3 Credit Scoring Methods: Latest Trends and Points to Consider

"Credit Scoring Methods: Latest Trends and Points to Consider" is a valuable contribution to the field of credit scoring research. Its comprehensive review, structured analysis, and focus on practical insights make it a valuable resource for researchers and practitioners alike. Addressing the limitations through further research, broader scope, and empirical validation would further strengthen the paper's impact and provide an even more comprehensive and insightful overview of the evolving landscape of credit scoring methods [9].

KeAi
CHINESE ROOTS
GLOBAL IMPACT

Available online at www.sciencedirect.com

ScienceDirect

The Journal of Finance and Data Science 8 (2022) 180–201

http://www.keaipublishing.com/en/journals/jfds/

Credit scoring methods: Latest trends and points to consider

Anton Markov*, Zinaida Seleznyova, Victor Lapshin [1]

National Research University Higher School of Economics, 20 Myasnitskaya Ulitsa, Moscow, 101000, Russia

Received 28 May 2022; accepted 26 July 2022
Available online 7 August 2022

Abstract

Credit risk is the most significant risk by impact for any bank and financial institution. Accurate credit risk assessment affects an organisation's balance sheet and income statement, since credit risk strategy determines pricing, and might even influence seemingly unrelated domains, e.g. marketing, and decision-making. This article aims at providing a systemic review of the most recent (2016–2021) articles, identifying trends in credit scoring using a fixed set of questions. The survey methodology and questionnaire align with previous similar research that analyses articles on credit scoring published in 1991–2015. We seek to compare our results with previous periods and highlight some of the recent best practices in the field that might be useful for future researchers.
© 2022 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

JEL classification: **G320** Financing Policy; Financial Risk and Risk Management; Capital and Ownership Structure; Value of Firms; Goodwill; **G210** Banks; Depository Institutions; Micro Finance Institutions; Mortgages; **C440** Operations Research; Statistical Decision Theory; **C650** Miscellaneous Mathematical Tools; **C830** Survey Methods; Sampling Methods; **C450** Neural Networks and Related Topics

Keywords: Credit scoring; Survey; Statistics; Machine learning; Data mining; Performance assessment

Contents

1. Introduction 181
2. Survey methodology 182

Figure 2.1.3 Credit Scoring Methods: Latest Trends and Points to Consider

2.2 Limitation of Previous Studies

Previous studies did not cover on the exhibits several weaknesses that collectively impact its overall effectiveness. Its simplicity, while user-friendly, may compromise accuracy by providing only a basic estimation, lacking the intricacies of credit scoring algorithms employed by credit bureaus. Relying on a limited set of financial data fields further restricts its ability to consider nuanced factors crucial in traditional credit scoring models, potentially leading to less comprehensive outcomes. The calculator's lack of customization options hinders its adaptability to diverse financial scenarios, impacting the precision of credit score estimates. Notably, the absence of sentiment analysis deprives the calculator of a holistic view of financial habits, potentially diminishing the accuracy of credit score predictions. Additionally, the project faces challenges in transparency, as it lacks detailed explanations of model choices, hindering users' comprehension and limiting its educational value. Potential overfitting concerns, without clear indications of mitigation strategies, raise questions about the model's reliability and applicability beyond the provided dataset. These weaknesses collectively highlight areas for improvement to enhance the calculator's accuracy, adaptability, and transparency. Besides that, previous studies did not implement the NLP sentiment analysis to resolve their problems.

CHAPTER 3 System Methodology/Approach

3.0 Methodology

Agile methodology is employed in projects like integrating sentiment analysis into credit risk prediction due to its adaptability and responsiveness. In dynamic and evolving endeavors, such as understanding user sentiments and predicting credit risk, the ability to accommodate changes and refine strategies over time is crucial. Agile's iterative approach allows for continuous development and regular stakeholder feedback, ensuring that the project stays aligned with evolving requirements and expectations.

Figure 3.0 Agile Methodology

Requirements:

Agile starts with a high-level understanding of project requirements of the credit scoring model with sentiment analysis. Initial requirements are identified, and as the project progresses, they are refined based on ongoing feedback and insights from stakeholders.

Design:

The design phase in Agile involves creating an initial design that aligns based on the set of requirements of credit scoring. Design iterations occur as the project evolves, accommodating new insights or changes identified during development. The design

phase also need to determine the hardware and software that needs to be use in the project.

Development:

During this phase, which is going to start developing the credit scoring model with sentiment analysis. The project going to use decompose to break down the complex task into a smaller task to ensure the task is manageable. This allows for the creation of increments of the system, fostering a continuous and iterative development process.

Testing:

Weekly testing is integrated into the development process to quickly and efficiently test new functionalities. Continuous testing practices are followed, ensuring that any issues are identified, addressed promptly during development, and the result came out have meet the objectives of the project.

Deployment:

Agile promotes frequent and incremental deployments. Small increments of the system are regularly released, ensuring that new features or improvements are quickly available. This phased deployment approach allows for adaptability and responsiveness.

Review:

Regular retrospectives are conducted at the end of each sprint to reflect on what went well and what could be improved. This feedback loop informs the next iterations, ensuring continuous improvement throughout the project.

3.1 System Design Diagram/Equation

3.1.1 System Architecture Diagram

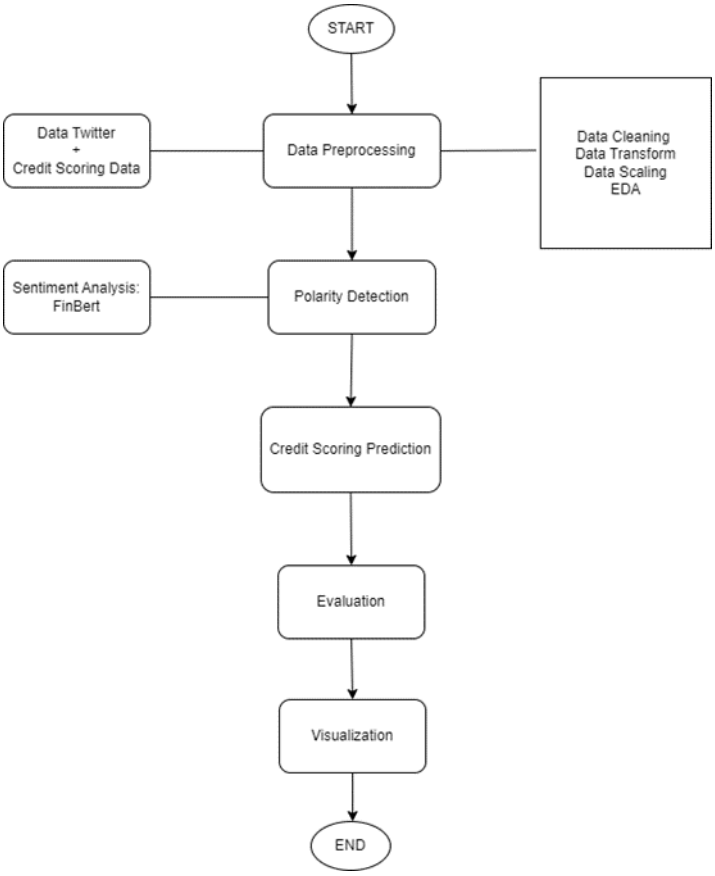


Figure 3.3.1 System Design

Phase 1: Data Preprocessing

In the Phase 1, the project needs to have a Twitter dataset and traditional loan credit scoring dataset to go through the data preprocessing phase to perform data cleaning because the datasets may contain the noise data like outliers, the duplicates, and error inputs. The data cleaning will help us to identify and fix all the anomalies to ensure that the quality and accuracy of the data during the training Machine Learning (ML). Besides that, data transform is also a necessary step in data preprocessing to ensure the data is suitable to perform analysis and modeling task. Lastly, data scaling and EDA is also important in the data preprocessing. In this phase, the project needs to import the dependencies during the data preprocessing.

Phase 2: Polarity Detection

In the Phase 2, after the data preprocessing, we need to perform sentiment analysis to detect the polarity of the text after lemmatized. The FinBert is pre-trained sentiment analysis in use of financial sector, so we choose FinBert to perform polarity detection to have more accurate model. The reason we choose FinBert is FinBert is a specialized version of Bert, specifically train on the financial text data and fine-tune for financial task.

Phase 3: Credit Scoring Prediction

In the Phase 3, we need to train the ML model to calculate the creditworthiness. The project will adopt 3-5 model like support vector machine (SVM), Neural network (NN) and more to train the model. The way we calculate the creditworthiness is the two datasets Twitter and credit scoring data is executed in parallel, the traditional credit scoring data which is consist 80% of the total creditworthiness score and the Twitter data will consist 20% of the total credit. For example, the applicant creditworthiness score needs at least 640 only will approve the loan application, but the applicant scores only have 600 so if the twitter sentiment analysis show positive then the positive polarity worth 40 score to let the applicant pass the application.

Phase 4: Evaluation

In the Phase 4, the evaluation is to assess the performance of all the trained model on the new or unseen data to see how good it generalizes to new data. The Algorithm that we use is around 3-5 Algorithm like RandomForest, Logistic Regression, Decision Tree and more. The evaluation can determine which model is suitable to fulfil the project objectives and comparing different model to select the most suitable one to build the most effectiveness of ML system.

Phase 6: Visualization

In the Phase 5, the visualization is to have some UI by using python flask to let the users to input their personal information like salary and the text data to get the text or commend in twitter platform. The web application is let the user easy to use to predict their credit scoring and let them know their majority sentiment analysis result from social media platform.

3.1.2 Use Case Diagram and Description

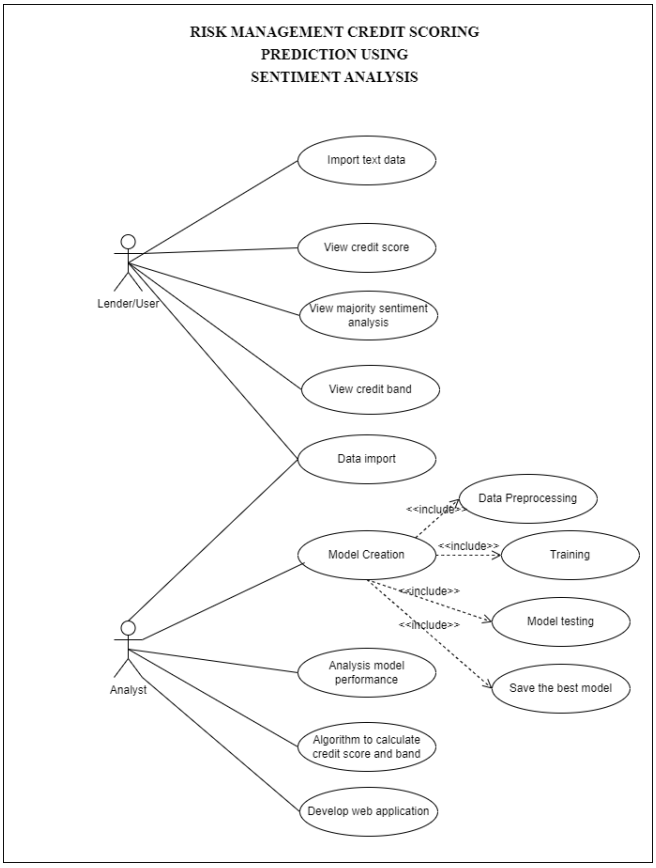


Figure 3.1.2.1 Use Case Diagram

This use case diagram depicts a risk management credit scoring system that leverages sentiment analysis for predictions. The system involves two primary actors: the Lender/User and the Analyst, each with distinct roles and capabilities. The Lender/User can import text data, view credit scores, access majority sentiment analysis results, and check credit band classifications. On the other hand, the Analyst has more technical responsibilities, including data import and model creation. The model creation process is particularly complex, involving data preprocessing, training, testing, and saving the best model. Additionally, the Analyst can analyse model performance, develop algorithms to calculate credit scores and bands, and create a web application for the system. This approach to credit scoring is innovative as it incorporates sentiment analysis alongside traditional financial metrics, potentially allowing for a more nuanced evaluation of credit risk. The system seems designed to cater to both end-users who need quick access to credit information and analysts who maintain and refine the underlying predictive models and algorithms.

3.1.3 Activity Diagram

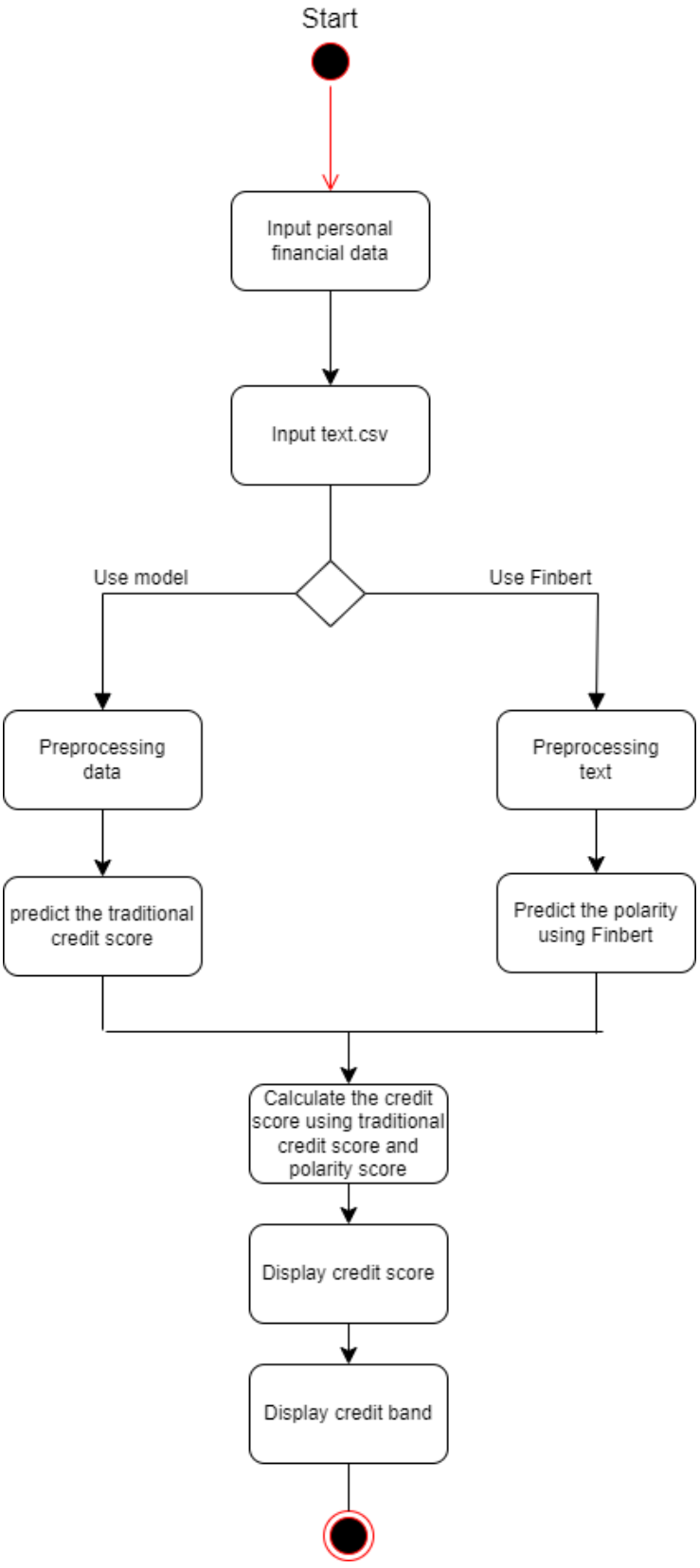


Figure 3.1.3 activity diagram

Chapter 4: System Design

In this chapter will outlines the design and structure of our credit scoring prediction system, which uses sentiment analysis to enhance accuracy. We start with a Block Diagram that shows how the main components interact. Next, we detail the specifications for each component, including both hardware and software. We also cover System Components Specifications needed for the system. Finally, we explain how all components work together to process data and generate predictions. This chapter provides a clear guide for understanding and recreating the system.

4.1 Sentiment Analysis

Figure 3.3.1.1 Sentiment Analysis

Data Collection:

The first stage is to data collection from twitter platform and traditional loan approval or user input to perform data preprocessing.

Data Preprocessing:

Once the data was collected, we need to do data preprocessing to prepare a good dataset in polarity detection. To process data preprocessing, we need to import the dependencies to perform some technique. First, we need to use stopwords to remove all the symbol and neutral word like he, she , it, & , !. It also need to remove the words starting with http or @ because inside the tweets have many this kind like hyperlink and mention function inside the text of the tweets. After that, we need to apply lemmatize the words it is because Lemmatization is the process of reducing words to their base or root form, known as the lemma, while still ensuring that the reduced form belongs to the language's dictionary. This is particularly useful in natural language processing (NLP) and text mining tasks where words need to be normalized for analysis or comparison. For example, the lemma of the word "running" is "run", and the lemma of "better" is "good".

After that, we need to perform data scaling and EDA on the structure data which is credit scoring dataset. The credit scoring dataset have two type of variable the continuous variable and category variable so have different way to do it. First, need to fill missing value with means and fill NaN value with mode and standard scale the variable. We can plot the distribution of continuous variables and remove outlier and update the data frame for each variable. We can show the boxplots before and after. For the category variables we need to apply one-hot encode the specified categorical variables.

Data Analysis:

After completing the data preprocessing stage, the system will advance to analyzing the data to gain a deeper understanding of the text. Furthermore, during this analysis phase, the system will consider the length of sentences, enhancing the overall sentiment analysis process.

Sentiment Classification:

In this stage, we're using a special model called FinBERT, designed for financial language. It's great at figuring out if the sentiment in text is neutral, positive,

or negative. Our setup takes text as input and gives out labels showing the sentiment. This sentiment analysis is a key part of our project, helping us get important insights from financial text that can guide decision-making in finance, investing, and market analysis.

Sentiment Score:

After completing the sentiment classification process, where the model calculates both positive and negative polarities, the next step is to derive a sentiment score. This score acts as the final metric for determining whether the data conveys a positive, neutral, or negative sentiment.

4.2 Modelling Block Diagram

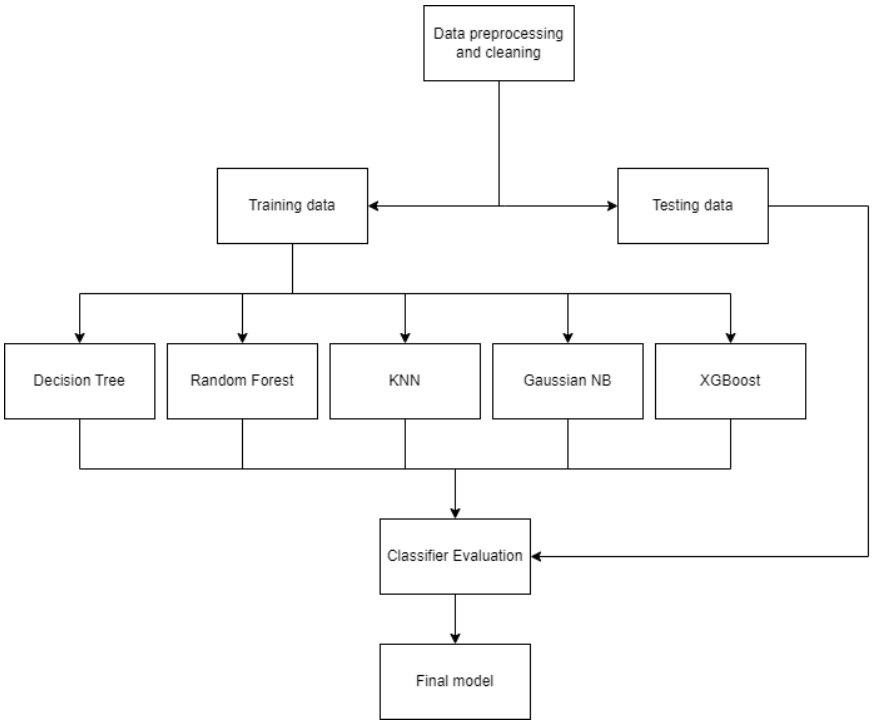


Figure 4.2.1 modelling

First, the project needs to perform data pre-processing and data cleaning for the credit score data to make sure it is ready to perform modelling. The dataset split it into 70% as training data and 30% to testing data. The 5 classifier which are Decision Tree, RandomForest, KNN, Gaussian NB, and XGBoost to compare the accuracy to know which one have the highest accuracy and it will be chosen as the classifier for the project.

4.3 Modelling flow

1. Data Loading

The process begins by loading the training data from a CSV file into a Pandas DataFrame. This step is crucial as it organizes the raw data into a structured format, making it easier to work with. By loading the data, this will establish the foundation for the entire analysis and modeling process.

2. Data Inspection

Once the data is loaded, the next step is to get a sense of what the project dealing with. The first step start by checking the basic details of the dataset, such as the number of rows and columns, the data types of each feature, and whether there are any missing values. Additionally, generate summary statistics for numerical features, such as the mean, median, and standard deviation. This initial exploration helps us understand the dataset's structure and highlights any potential issues that need addressing before proceed further.

3. Data Preprocessing

Data preprocessing is where we clean and prepare the data for modelling. This step often involves filling in or removing missing values, encoding categorical variables into a numerical format, and possibly creating new features from the existing ones. It might also normalize or standardize the data, especially if using models sensitive to feature scaling. The goal here is to ensure the data is clean and in the right format for training our machine learning models.

4. Model Training

With the data ready, move on to training various machine learning models. Common choices for credit score prediction include Decision Trees, Random Forests, Logistic Regression, Naive Bayes, K-Nearest Neighbors, and XGBoost. During this step, each model learns from the training data by identifying patterns and relationships between the input features and the target variable (credit score). The models are trained to minimize errors and improve their ability to make accurate predictions.

5. Model Evaluation

After training, it's important to assess how well each model is performing. To evaluate the models using metrics such as accuracy, precision, recall, and confusion matrix. These metrics give us insights into the strengths and weaknesses of each model, helping us understand how effectively they can distinguish between different credit score categories. This evaluation is critical for deciding which model will be the most reliable for making predictions.

6. Model Selection

Based on the evaluation results, will select the best-performing model. This involves comparing the performance of all the trained models and choosing the one that offers the best balance of accuracy, precision, recall, and other relevant metrics. The selected model is deemed the most suitable for predicting credit scores in new data.

7. Prediction

With the best model in hand, we can now use it to make predictions on new, unseen data. This might involve predicting credit scores for new applicants or updating scores for existing customers based on recent information. The model applies the patterns it learned during training to make these predictions.

8. Saving the Model

Finally, save the trained model for future use. By saving the model, the project can quickly load it and use it in production without needing to retrain it. This step is crucial for deploying the model in real-world applications, ensuring that predictions can be made efficiently as new data becomes available.

4.4 Weight of credit score and polarity

In the project, the traditional credit scoring methods is allocate 80% and 20% for sentiment analysis. The reason that set this distribution is traditional credit scoring have been proven that is more reliable in creditworthiness. recognizing the growing importance of alternative data sources, I've also incorporated sentiment analysis. By analyzing textual data, such as customer reviews or social media posts, sentiment analysis can offer additional insights that may capture nuances not reflected in traditional metrics. This distribution can approach the strengths of both traditional and modern to have a comprehensive credit assessment.

4.5 Credit band

The project have indicate the credit band based on the credit score after combine the sentiment analysis. The credit band will categorize into various bands to represent the quality of creditworthiness by evaluating the score to identify the predefined range and respective label for user. If the score is below 360, it is classified as "Poor," indicating significant risk. Scores ranging from 360 to 479 are labeled "Fair," suggesting some potential concerns but not as severe. A score between 480 and 599 is deemed "Good," representing a solid credit standing. Scores from 600 to 679 are categorized as "Very Good," reflecting a high level of credit reliability. Finally, scores of 680 and above are considered "Excellent," denoting exemplary creditworthiness. This tiered classification helps in quickly understanding and interpreting credit score levels.

4.6 Type of classifier

Here's the five types of machine learning algorithm that we use in this project. Each type of machine learning have different way to work and below will explain how it all works.

1.) Decision Tree

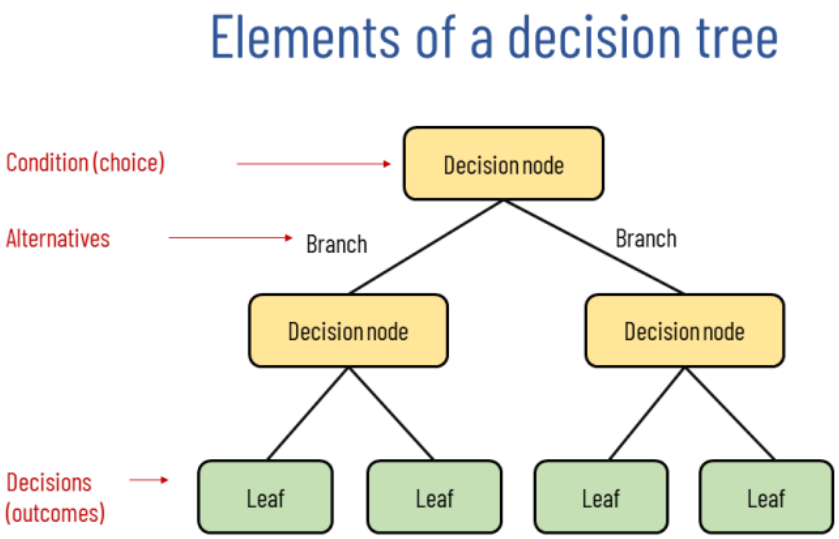


Figure 4.6.1 decision tree

A Decision Tree is a model that splits data into subsets based on feature values to create a tree-like structure. Each node in the tree represents a decision point based on a particular feature, and branches indicate the possible outcomes of that decision. The process continues until the data is split into subsets that are as homogeneous as possible with respect to the target variable. This model is intuitive and easy to visualize, making it a popular choice for both classification and regression tasks. Its simplicity allows for straightforward interpretation of how decisions are made.

2.) Random Forest

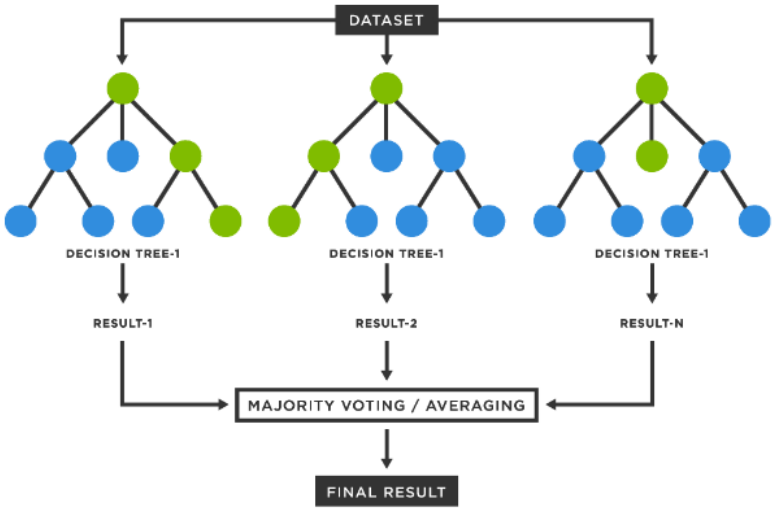


Figure 4.6.2 Random Forest

Random Forest is an ensemble learning method that enhances the performance of Decision Trees by aggregating the results of multiple trees. It creates a "forest" of Decision Trees, each trained on a random subset of the data and features. During prediction, the Random Forest combines the predictions from all individual trees—either by averaging (for regression) or by majority voting (for classification)—to produce a final result. This approach reduces the risk of overfitting and generally improves the accuracy and robustness of the model compared to using a single Decision Tree.

3.) KNN

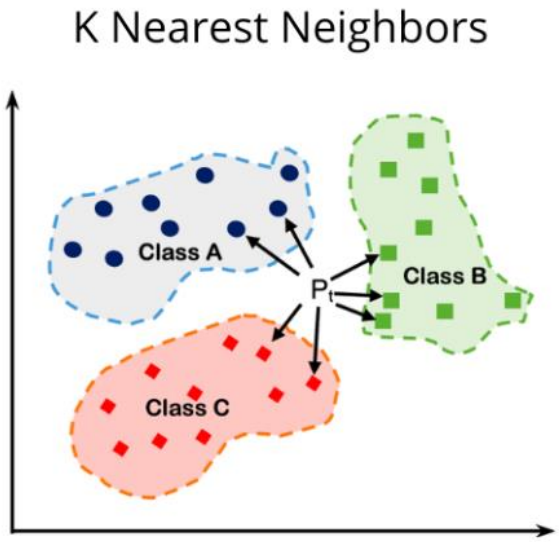


Figure 4.6.3 KNN

K-Nearest Neighbors is a simple, instance-based learning algorithm used for classification and regression. It works by finding the 'k' closest training examples to a given test point based on a distance metric, such as Euclidean distance. The output is determined by the majority class (for classification) or the average of the values (for regression) of these nearest neighbors. KNN does not require explicit training, as it makes decisions based on the entire dataset at prediction time, which can make it computationally expensive for large datasets but very flexible and easy to implement.

4.) Gaussian Naive Bayes

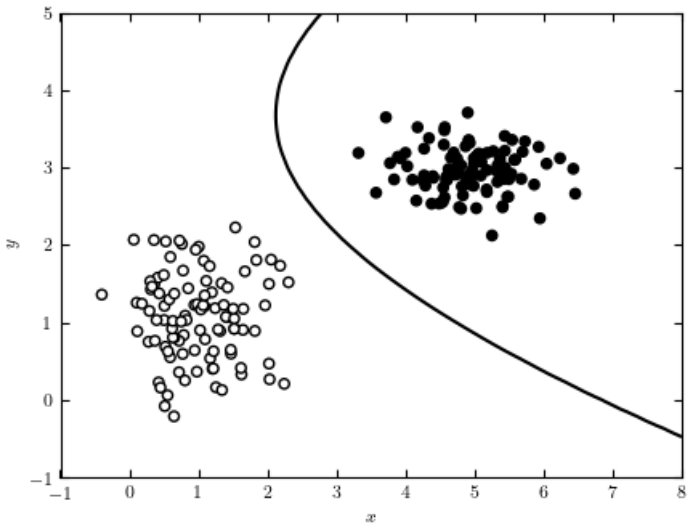


Figure 4.6.4 Gaussian Naive Bayes

Gaussian Naive Bayes is a probabilistic classifier based on Bayes' Theorem, with the assumption that features are conditionally independent given the class label. It specifically assumes that the features follow a Gaussian (normal) distribution, which allows it to compute probabilities based on the mean and variance of each feature for each class. This model is particularly efficient for large datasets and performs well even when the assumption of feature independence is not entirely accurate. Its simplicity and ease of implementation make it a popular choice for text classification and other tasks involving large feature spaces.

5.) XGBoost (Extreme Gradient Boosting)

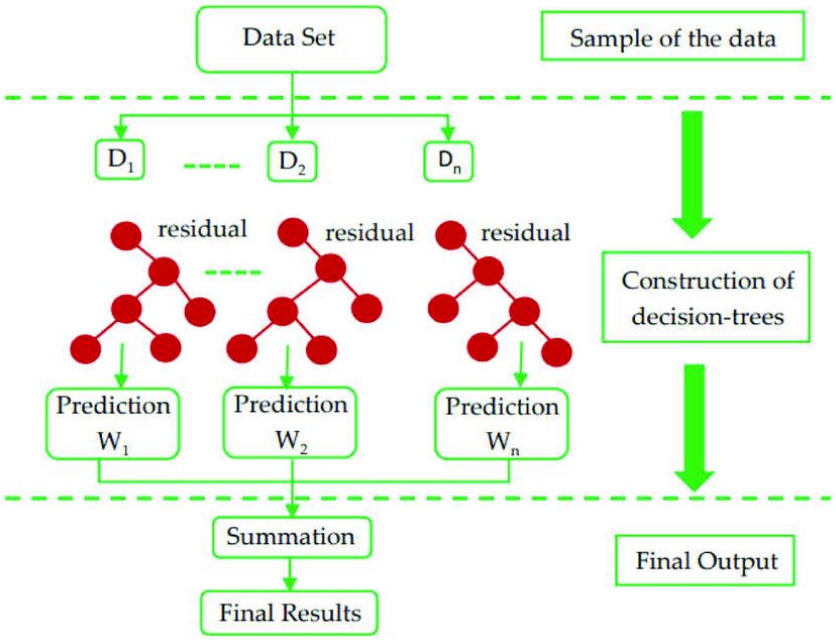


Figure 4.6.5 XGBoost

XGBoost is a powerful and flexible gradient boosting algorithm used for both classification and regression tasks. It builds an ensemble of decision trees sequentially, where each new tree corrects the errors of the previous ones by focusing on the residuals. XGBoost incorporates regularization to prevent overfitting, and it optimizes the training process through techniques like parallelization and tree pruning. Its ability to handle various types of data and its high performance in competitions make it a favoured choice for complex and large-scale machine learning problems.

Chapter 5: System Implementation

5.1 Hardware Setup

For my project, I'm working on an HP Omen Series laptop, which is equipped with an Intel Core i7-7700HQ processor. The laptop runs on Windows 10, providing a stable environment for all tasks. It has an NVIDIA GeForce GTX 1050 graphics card, which handles graphics-intensive applications smoothly. The system also includes 8GB of DDR4 RAM, allowing for efficient multitasking, and a 1TB PCIE SSD, which offers plenty of storage space and fast data access, making it a reliable setup for all my computing needs.

Description	Specifications
Model	HP Omen Series
Processor	Intel(R) Core(TM) i7-7700HQ
Operating System	Windows 10
Graphic	NVIDIA GeForce GTX 1050
Memory	8GB DDR4 RAM
Storage	1TB PCIE SSD

Table 5.1 Specifications of laptop

5.2 Software Setup

1.) Jupyter Notebook

Figure 5.2.1 jupyter notebook

Jupyter Notebook is an open-source, web-based interactive computing environment that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It is widely used for data analysis, scientific research, machine learning, and more.

2.) Visual Studio Code

Figure 5.2.2 VS code

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft, designed for building and debugging modern web and cloud applications.

3.) Flask Python

Figure 5.2.3 Flask

Flask is a lightweight and flexible web framework for Python, designed to help developers build web applications quickly and with minimal overhead.

4.) FinBert

FinBERT is a specialized variant of the BERT (Bidirectional Encoder Representations from Transformers) model, tailored specifically for financial text analysis tasks. Developed by researchers and practitioners in the field of natural language processing (NLP), FinBERT is trained on a large corpus of financial text data, including news articles, earnings call transcripts, and social media posts related to finance. Unlike general-purpose BERT models, FinBERT is fine-tuned using domain-specific datasets and objectives, enabling it to capture the nuances and intricacies of financial language more effectively. This domain-specific training allows FinBERT to generate contextualized word embeddings and understand the unique vocabulary and

semantics of financial jargon, making it well-suited for applications such as sentiment analysis, financial news classification, and stock price prediction. FinBERT has gained popularity in the finance industry for its ability to extract valuable insights from unstructured financial text data, helping analysts, traders, and investors make more informed decisions in the dynamic and complex world of finance.

Choosing Python with the Natural Language Toolkit (NLTK) for this project makes things easier. Python is a widely used language, and NLTK is specifically designed for working with language data. Together, they offer a straightforward way to analyze and understand the sentiments in text. Python's simplicity and NLTK's ready-to-use tools save time in developing the sentiment analysis part of the project. Plus, many people use Python and NLTK, so there's plenty of community support and updates. It's a practical choice for making the credit risk prediction process more effective and user-friendly [10].

5.3 Setting and Configuration

Before commencing the development process, it is essential to install several software and plugins on the laptop.

1.) Visual Studio Code
2.) Python
3.) Python Flask
4.) Jupyter notebook
5.) FinBert

5.4 Data Preprocessing

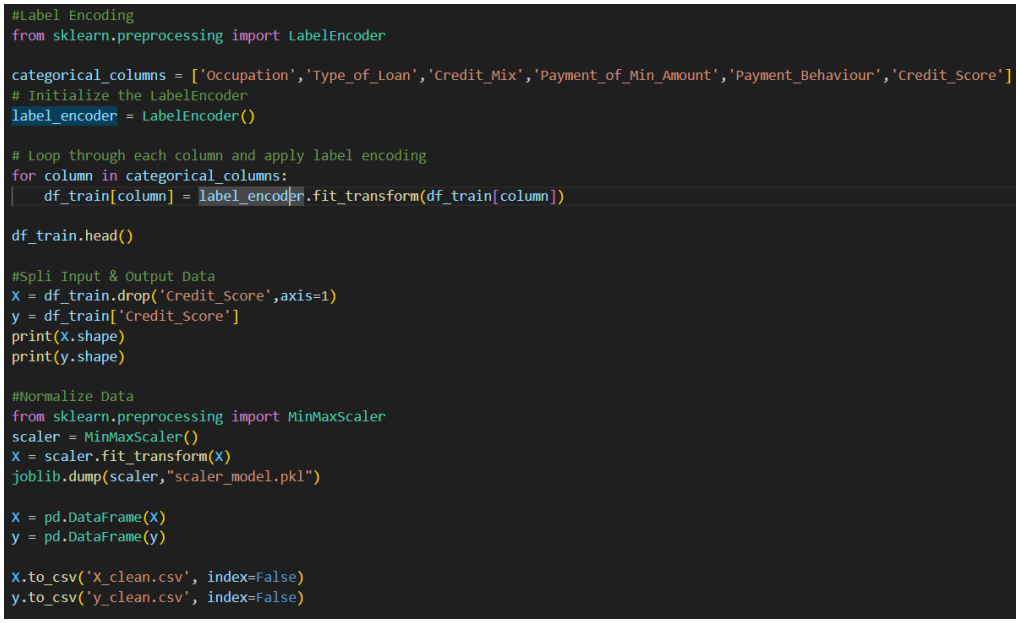
```python
def get_column_details(df,column):
    print("Details of",column,"column")

    #DataType of column
    print("\nDataType: ",df[column].dtype)

    #Check if null values are present
    count_null = df[column].isnull().sum()
    if count_null==0:
        print("\nThere are no null values")
    elif count_null>0:
        print("\nThere are ",count_null," null values")

    #Get Number of Unique Values
    print("\nNumber of Unique Values: ",df[column].nunique())

    #Get Distribution of Column
    print("\nDistribution of column:\n")
    print(df[column].value_counts())

def fill_missing_with_group_mode(df, groupby, column):
    print("\nNo. of missing values before filling with group mode:",df[column].isnull().sum())

    # Fill with local mode
    mode_per_group = df.groupby(groupby)[column].transform(lambda x: x.mode().iat[0])
    df[column] = df[column].fillna(mode_per_group)

    print("\nNo. of missing values after filling with group mode:",df[column].isnull().sum())
```

```python
#Method to clean categorical field

def clean_categorical_field(df,groupby,column,replace_value=None):
    print("\n-----------------------------------------------------------")
    print("\nCleaning steps ")

    #Replace with np.nan
    if replace_value!=None:
        df[column] = df[column].replace(replace_value,np.nan)
        print(f"\nGarbage value {replace_value} is replaced with np.nan")

    #For each Customer_ID, assign same value for the column
    fill_missing_with_group_mode(df,groupby,column)
```

```python
# Handle Outliers and null values
def fix_inconsistent_values(df, groupby, column):
    print("\nExisting Min, Max Values:", df[column].apply([min, max]), sep='\n', end='\n')

    df_dropped = df[df[column].notna()].groupby(groupby)[column].apply(list)

    # Calculate mode for each group
    mode_values = df_dropped.apply(lambda x: stats.mode(x, keepdims=True))

    # Extract min and max values, handling potential empty results
    mini = mode_values.apply(lambda x: x.mode[0] if x.mode.size > 0 else np.nan).min()
    maxi = mode_values.apply(lambda x: x.mode[0] if x.mode.size > 0 else np.nan).max()

    if np.isnan(mini) or np.isnan(maxi):
        print(f"Warning: Unable to determine valid min/max for {column}. Using original column min/max.")
        mini, maxi = df[column].min(), df[column].max()

    # Assign Wrong Values to NaN
    col = df[column].apply(lambda x: np.nan if ((x < mini) | (x > maxi) | (x < 0)) else x)

    # Fill with local mode
    mode_by_group = df.groupby(groupby)[column].transform(lambda x: x.mode()[0] if not x.mode().empty else np.nan)
    df[column] = col.fillna(mode_by_group)
    df[column].fillna(df[column].mean(), inplace=True)

    print("\nAfter Cleaning Min, Max Values:", df[column].apply([min, max]), sep='\n', end='\n')
    print("\nNo. of Unique values after Cleaning:", df[column].nunique())
    print("\nNo. of Null values after Cleaning:", df[column].isnull().sum())
```

Figure 5.4.1 function to perform cleaning

These functions collectively handle various aspects of data preprocessing, including cleaning categorical and numerical fields, handling missing values, and visualizing data distributions. The methods ensure that the dataset is prepared for modeling by addressing inconsistencies, missing data, and providing insightful visual representations of the data.



```
#Label Encoding
from sklearn.preprocessing import LabelEncoder

categorical_columns = ['Occupation','Type_of_Loan','Credit_Mix','Payment_of_Min_Amount','Payment_Behaviour','Credit_Score']
# Initialize the LabelEncoder
label_encoder = LabelEncoder()

# Loop through each column and apply label encoding
for column in categorical_columns:
    df_train[column] = label_encoder.fit_transform(df_train[column])

df_train.head()

#Spli Input & Output Data
X = df_train.drop('Credit_Score',axis=1)
y = df_train['Credit_Score']
print(X.shape)
print(y.shape)

#Normalize Data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X = scaler.fit_transform(X)
joblib.dump(scaler,"scaler_model.pkl")

X = pd.DataFrame(X)
y = pd.DataFrame(y)

X.to_csv('X_clean.csv', index=False)
y.to_csv('y_clean.csv', index=False)
```

Figure 5.4.2 Label Encoding, Normalize Data

The figure above shown a series of preprocessing steps for a machine learning pipeline, starting with label encoding and ending with data normalization and saving. First, the LabelEncoder from sklearn.preprocessing is used to convert categorical variables into numerical labels, which is necessary for many machine learning algorithms that require numerical input. The code applies label encoding to several categorical columns in the df_train DataFrame.

Next, the code splits the data into features (X) and the target variable (y), with X containing all columns except the target variable 'Credit_Score' and y holding the 'Credit_Score'. It then normalizes the feature data using MinMaxScaler, which scales the data to a range between 0 and 1, making it easier for machine learning algorithms to process. The scaler is saved to a file for future use, ensuring that the same scaling transformation can be applied to new data consistently. Finally, both the normalized features and the target variable are saved to CSV files for further analysis or model training. This preprocessing pipeline helps prepare the data for efficient and effective model training.

```
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
# Initialize lemmatizer
lemmatizer = WordNetLemmatizer()

def load_custom_stopwords(file_path='custom_stopwords.txt'):
    with open(file_path, 'r') as f:
        custom_stopwords = set(f.read().splitlines())
    return custom_stopwords

# Check if the custom stop words file exists
if os.path.exists('custom_stopwords.txt'):
    # Load custom stop words from file
    stop_words = load_custom_stopwords()
else:
    # Download stop words and customize them
    stop_words = set(stopwords.words('english'))
    stop_words.discard('no')
    stop_words.discard('not')
    stop_words.update(['lol', 'awww', 'idk', 'im', 'otw', 'ive', 'asap', 'btw'])

    # Save the customized stop words to a file
    with open('custom_stopwords.txt', 'w') as f:
        for word in stop_words:
            f.write(f"{word}\n")
```

Figure 5.4.3 text cleaning

```
def get_wordnet_pos(word):
    return nltk.corpus.wordnet.VERB

def preprocessing(text):
    # Remove non-alphabetic characters (except '@') and convert to lower case
    text = re.sub(r'[^A-Za-z\s@]', '', text).lower()
    # Tokenize the text to ensure proper lemmatizing
    #words = nltk.word_tokenize(text)
    words = text.split()
    # Remove words starting with 'http' or '@'
    words = [word for word in words if not word.startswith('http') and not word.startswith('@')]
    # Remove stop words and lemmatize the words
    return ' '.join([lemmatizer.lemmatize(word, get_wordnet_pos(word)) for word in words if word not in stop_words])

finbert = BertForSequenceClassification.from_pretrained('yiyanghkust/finbert-tone',num_labels=3)
tokenizer = BertTokenizer.from_pretrained('yiyanghkust/finbert-tone')
```

Figure 5.4.4 text preprocessing

Text preprocessing pipeline using the Natural Language Toolkit (NLTK) for text data. It starts by downloading necessary NLTK resources such as tokenizers, WordNet for lemmatization, and part-of-speech taggers. A custom stopwords file is used if it exists, containing user-defined stopwords, otherwise, a default list of English stopwords is customized by removing some words and adding new ones, and then saved to a file for future use. The preprocessing function takes a text input, removes non-alphabetic characters (except '@'), and converts the text to lowercase. It then tokenizes the text by splitting it into words, removes any words starting with 'http' or '@' (typically URLs or mentions), and filters out stopwords. The remaining words are lemmatized using WordNet to reduce them to their base forms, and the processed words are joined back into a single string. This preprocessing step cleans and standardizes text data, preparing it for further analysis or modeling.

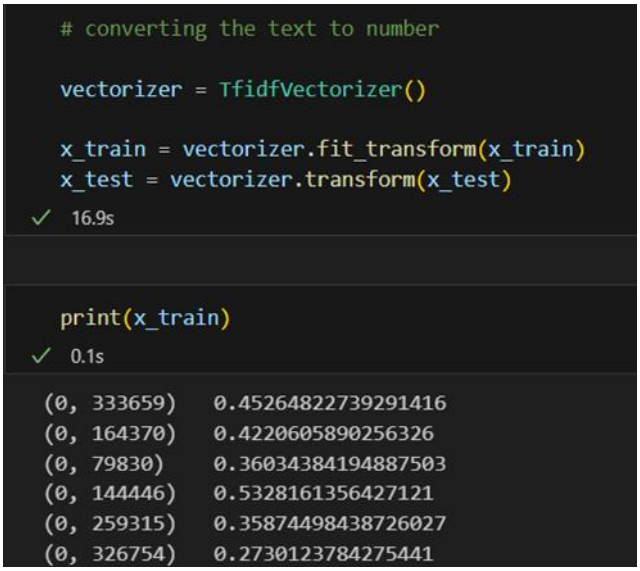
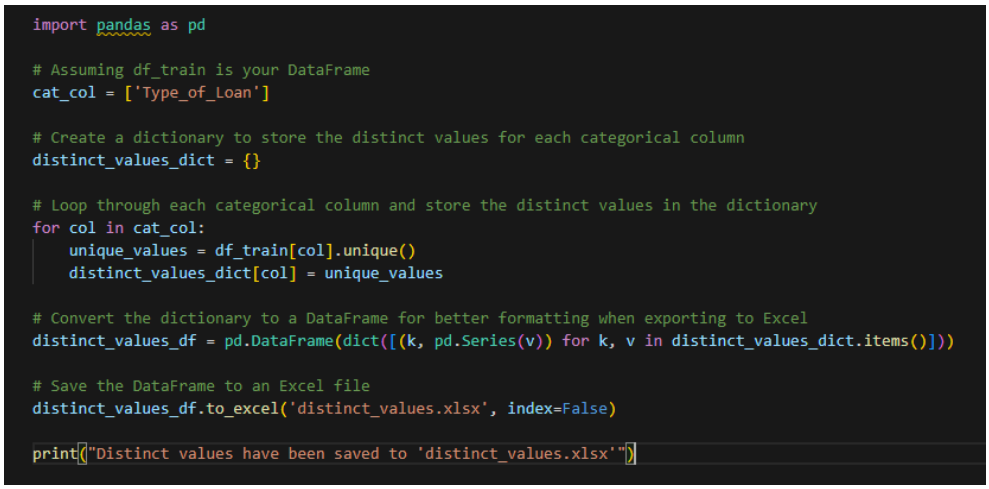


Figure 5.4.5 Using vectorizer

	text	lemmatized_text
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	thats bummer shoulda get david carr third day
1	is upset that he can't update his Facebook by ...	upset cant update facebook texting might cry r...
2	@Kenichan I dived many times for the ball. Man...	dive many time ball manage save rest go bound
3	my whole body feels itchy and like its on fire	whole body feel itchy like fire
4	@nationwideclass no, it's not behaving at all....	no not behave mad cant see
5	@Kwesidei not the whole crew	not whole crew
6	Need a hug	need hug
7	@LOLTrish hey long time no see! Yes.. Rains a...	hey long time no see yes rain bite bite fine t...

Figure 5.4.6 Lemmatized Text

5.4.1 Data Transformation



```python
import pandas as pd

# Assuming df_train is your DataFrame
cat_col = ['Type_of_Loan']

# Create a dictionary to store the distinct values for each categorical column
distinct_values_dict = {}

# Loop through each categorical column and store the distinct values in the dictionary
for col in cat_col:
    unique_values = df_train[col].unique()
    distinct_values_dict[col] = unique_values

# Convert the dictionary to a DataFrame for better formatting when exporting to Excel
distinct_values_df = pd.DataFrame(dict([(k, pd.Series(v)) for k, v in distinct_values_dict.items()]))

# Save the DataFrame to an Excel file
distinct_values_df.to_excel('distinct_values.xlsx', index=False)

print("Distinct values have been saved to 'distinct_values.xlsx'")
```

Figure 5.4.1.1 data transform

The provided Python script is designed to process categorical data from a DataFrame and save the distinct values into an Excel file. It begins by importing the pandas library, essential for data manipulation, under the alias pd. The script assumes that a DataFrame named df_train already exists and contains the data to be processed. It then defines a list cat_col, which holds the name of the categorical column of interest, in this case, 'Type_of_Loan'. An empty dictionary called distinct_values_dict is initialized to store the unique values found within this column. The script proceeds by looping through each column specified in the cat_col list, though in this instance, it only deals with the 'Type_of_Loan' column. Within the loop, it extracts the unique values from the column using df_train[col].unique() and stores these values in the dictionary, using the column name as the key. Once all unique values are collected, the dictionary is converted into a new DataFrame, distinct_values_df, which allows for better formatting when exporting the data to Excel. The script then saves this new DataFrame to an Excel file named 'distinct_values.xlsx' and prints a confirmation message to indicate that the file has been successfully saved.

```
#Drop columns
print("Size of Dataset before dropping columns : ",df_train.shape)
drop_columns = ['ID','Customer_ID','Name','SSN','Month']
df_train.drop(drop_columns,axis=1,inplace=True)
print("Size of Dataset after dropping columns : ",df_train.shape)
df_train.to_csv('df_train_clean.csv', index=False)
```

```
Size of Dataset before dropping columns :  (100000, 28)
Size of Dataset after dropping columns :  (100000, 23)
```

Figure 5.4.1.2 Drop columns

The columns that need to drop is ID, Customer ID, Name, SSN, and Month because these columns are irrelevant, so we just drop it and the size of dataset after dropping columns is 22

```
# Initialize the LabelEncoder
label_encoder = LabelEncoder()
categorical_columns = ['Occupation','Type_of_Loan','Credit_Mix','Payment_of_Min_Amount','Payment_Behaviour','Credit_Score']
# Loop through each column and apply label encoding
for column in categorical_columns:
    df_train[column] = label_encoder.fit_transform(df_train[column])

df_train.head()
```

Figure 5.4.1.3 LabelEncoder

The Figure above demonstrates how to convert categorical data into numerical format using LabelEncoder from the sklearn library. This process, known as label encoding, is a common preprocessing step in machine learning, especially when dealing with categorical features that need to be converted into a format that machine learning algorithms can understand.

First, the LabelEncoder class is initialized by creating an instance named label_encoder. The script then defines a list called categorical_columns, which contains the names of the categorical columns in the DataFrame df_train. These columns include 'Occupation', 'Type_of_Loan', 'Credit_Mix', 'Payment_of_Min_Amount', 'Payment_Behaviour', and 'Credit_Score'.

Next, a loop is employed to iterate through each column name in the categorical_columns list. For each column, the script applies the label encoding by calling label_encoder.fit_transform(df_train[column]). This method transforms each unique category within the column into a corresponding numerical label. The transformed data replaces the original categorical data in the df_train DataFrame.

Finally, the script displays the first few rows of the transformed DataFrame using df_train.head(). This allows you to see how the categorical data has been converted into numerical format, making it ready for further analysis or modeling in machine learning projects. This step is crucial for preparing the data for algorithms that require numerical input, ensuring that categorical variables are appropriately represented.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 28 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   ID                        100000 non-null  object
 1   Customer_ID               100000 non-null  object
 2   Month                     100000 non-null  object
 3   Name                      90015 non-null   object
 4   Age                       100000 non-null  object
 5   SSN                       100000 non-null  object
 6   Occupation                100000 non-null  object
 7   Annual_Income             100000 non-null  object
 8   Monthly_Inhand_Salary     84998 non-null   float64
 9   Num_Bank_Accounts         100000 non-null  int64
 10  Num_Credit_Card           100000 non-null  int64
 11  Interest_Rate             100000 non-null  int64
 12  Num_of_Loan               100000 non-null  object
 13  Type_of_Loan              88592 non-null   object
 14  Delay_from_due_date       100000 non-null  int64
 15  Num_of_Delayed_Payment    92998 non-null   object
 16  Changed_Credit_Limit      100000 non-null  object
 17  Num_Credit_Inquiries      98035 non-null   float64
 18  Credit_Mix                100000 non-null  object
 19  Outstanding_Debt          100000 non-null  object
...
 26  Monthly_Balance           98800 non-null   object
 27  Credit_Score              100000 non-null  object
dtypes: float64(4), int64(4), object(20)
```

Figure 5.4.1.4 Check for Null

As the Figure 4.3.1 shown as above, the dataset appears some of the columns are null. The dataset has continuous variables and categorical variables split it out because different variables have different method.

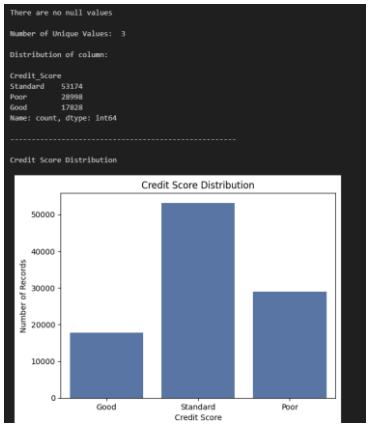


Figure 5.4.1.5credit score

The figure shown that the dataset have 100000 distinct record with no null values present.

5.4.2 Data Visualization

For data visualization, it only will show some of categorical variables and continuous variables due to the data have too many variables.

categorical variables

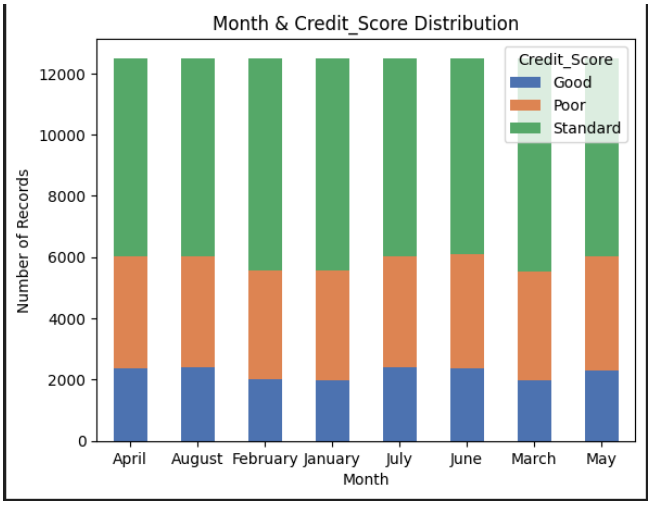


Figure 5.4.2.1 month and credit score distribution

The figure shown that January to August have same equal distribution 12500 and there is no null. Every variable we need to make sure there is no null.

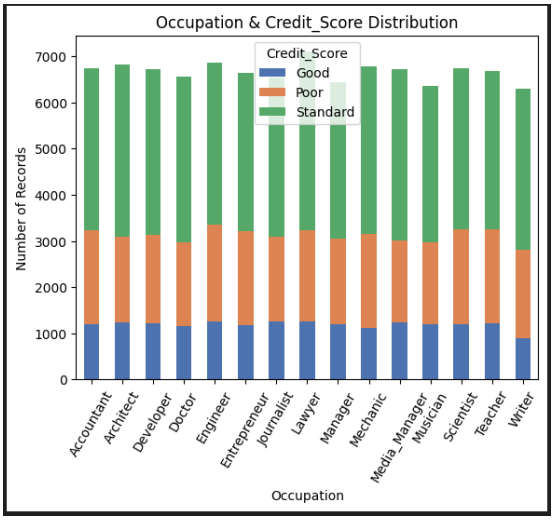


Figure 5.4.2.2 Occupation and credit score distribution

Figure shown above have 16 of unique values which are Accountant, Architect, Developer, Doctor, Engineer, Entrepreneur, Journalist, Lawyer, Manager, Mechanic, Media Manager, Musician, Scientist, Teacher, and Writer.

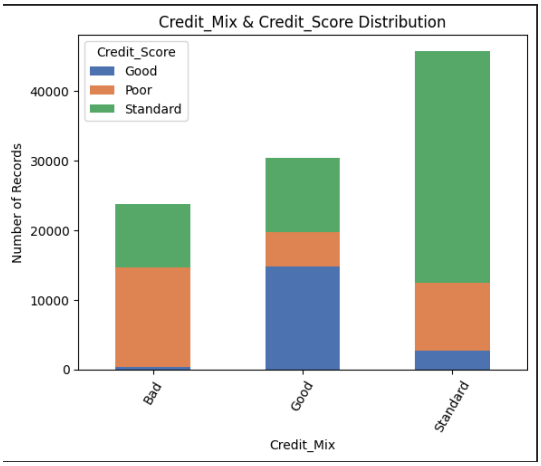


Figure 5.4.2.3 credit mix and credit score distribute

The figure above shown as illustrates the distribution of credit scores across different credit mix categories: "Bad," "Good," and "Standard." The "Standard" credit mix has the highest number of records, with the majority of these being "Standard" and "Good" credit scores, and a smaller portion of "Poor" scores. In contrast, the "Bad" credit mix is predominantly associated with "Poor" credit scores, with very few "Good" scores. The "Good" credit mix shows a more balanced distribution between "Good" and "Standard" scores, with a smaller fraction of "Poor" scores. This distribution suggests a strong relationship between credit mix and credit score, where individuals with a "Standard" credit mix are more likely to have better credit scores, while those with a "Bad" mix are more likely to have poorer scores. These trends indicate that credit mix could be a significant factor in predicting credit scores, with clear patterns emerging across different categories.

Continuous variables

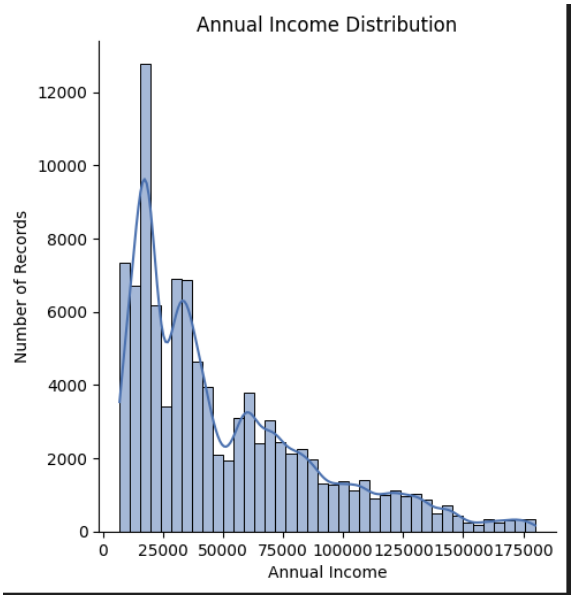


Figure 5.4.2.4 Income distribute

The chart displays the distribution of annual income across a dataset, with the x-axis representing income levels and the y-axis showing the number of records. The distribution is heavily skewed towards lower income levels, with the highest concentration of records falling in the 0 to 25,000 range. There is a noticeable peak around 20,000, indicating that a significant portion of the dataset falls within this income bracket. As income increases, the number of records steadily declines, with fewer records found in higher income brackets. The distribution shows a long tail extending towards higher income levels, but the frequency of records decreases sharply after 50,000.

5.5 FINBERT

```
finbert = BertForSequenceClassification.from_pretrained('yiyanghkust/finbert-tone',num_labels=3)
tokenizer = BertTokenizer.from_pretrained('yiyanghkust/finbert-tone')
#nlp = pipeline("sentiment-analysis", model='finiteautomata/bertweet-base-sentiment-analysis', token


#results=analyzer.predict(preprocessing(text))
nlp = pipeline("sentiment-analysis", model=finbert, tokenizer=tokenizer)
results = nlp(preprocessing(text))
return results  #LABEL_0: neutral; LABEL_1: positive; LABEL_2: negative
```

Figure 5.5.1 FINBERT

FinBERT is a specialized version of the BERT model tailored for financial sentiment analysis. It builds on BERT's transformer architecture, which enables it to process text bidirectionally, considering both preceding and following words to understand context more deeply. Initially, FinBERT is pre-trained on a large text corpus to learn general language patterns. It is then fine-tuned on financial texts, such as earnings reports and financial news, to grasp financial terminology and sentiment nuances. This fine-tuning allows FinBERT to effectively classify the sentiment of financial content into categories like positive, negative, or neutral. By using the Hugging Face transformers library, can easily load FinBERT, preprocess your text data, and obtain sentiment predictions, streamlining the integration of this powerful model into the applications.

5.6 Model Training

```
from sklearn.model_selection import train_test_split
import statistics
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
import xgboost as xgb
from sklearn.metrics import accuracy_score, precision_score, recall_score,classification_report,confusion_matrix
from joblib import dump, load
```

Figure 5.6.1 Import library

The figure shown the necessary library for the model training.

```
#Method to evaluate the performance of the model
def evaluate_model(y_test,y_pred):
    print("Classification Report")
    print(classification_report(y_test, y_pred))

    print("\n----------------------------------------------\n")
    # Compute confusion matrix
    cm = confusion_matrix(y_test, y_pred)

    # Create a heatmap of the confusion matrix using Seaborn
    sns.heatmap(cm, annot=True, cmap='Greens',fmt='.0f')

    plt.xlabel('Predicted Labels')
    plt.ylabel('True Labels')
    plt.title('Confusion Matrix')

    plt.show()
```

Figure 5.6.2 function to evaluate model

Here is the function to evaluate the performance of models.

```python
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint, uniform
# Define parameter distributions for each classifier
param_dists = {
    'Decision Tree': {
        'max_depth': randint(1, 32),
        'min_samples_split': randint(2, 20),
        'min_samples_leaf': randint(1, 20)
    },
    'Random Forest': {
        'n_estimators': randint(100, 150),
        'max_depth': randint(5, 50),
        'min_samples_split': randint(2, 20),
        'min_samples_leaf': randint(1, 10),
        'max_features': ['sqrt', 'log2']
    },
    'KNN': {
        'n_neighbors': randint(1, 20),
        'weights': ['uniform', 'distance'],
        'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']
    },
    'Gaussian NB': {
        'var_smoothing': uniform(1e-10, 1e-8)
    },
    'XGBoost': {
        'n_estimators': randint(100, 1000),
        'max_depth': randint(3, 10),
        'learning_rate': uniform(0.01, 0.3),
        'subsample': uniform(0.6, 0.4),
        'colsample_bytree': uniform(0.6, 0.4)
    }
}
```

Figure 5.6.3 Define parameter distribution

The figure above is used to define parameter distributions for hyperparameter tuning of various machine learning classifiers using RandomizedSearchCV from sklearn. This technique helps in finding the optimal hyperparameters by randomly sampling from predefined distributions rather than exhaustively searching through all possible combinations.

```
Performing RandomizedSearchCV for Decision Tree
Fitting 3 folds for each of 50 candidates, totalling 150 fits
Best parameters for Decision Tree: {'max_depth': 25, 'min_samples_leaf': 14, 'min_samples_split': 19}
Best cross-validation score: 0.7262

Performing RandomizedSearchCV for Random Forest
Fitting 3 folds for each of 50 candidates, totalling 150 fits
Best parameters for Random Forest: {'max_depth': 29, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 126}
Best cross-validation score: 0.7994

Performing RandomizedSearchCV for KNN
Fitting 3 folds for each of 50 candidates, totalling 150 fits
Best parameters for KNN: {'algorithm': 'brute', 'n_neighbors': 6, 'weights': 'distance'}
Best cross-validation score: 0.7308

Performing RandomizedSearchCV for Gaussian NB
Fitting 3 folds for each of 50 candidates, totalling 150 fits
Best parameters for Gaussian NB: {'var_smoothing': 3.845401188473625e-09}
Best cross-validation score: 0.6391

Performing RandomizedSearchCV for XGBoost
Fitting 3 folds for each of 50 candidates, totalling 150 fits
Best parameters for XGBoost: {'colsample_bytree': 0.9022204554172195, 'learning_rate': 0.07863944964748673, 'max_depth': 9, 'n_estimators': 798, '
ple': 0.8779139732158818}
Best cross-validation score: 0.7944
```

Figure 5.6.4 Model comparison

The figure shows the results of performing hyperparameter tuning using RandomizedSearchCV for several machine learning models: Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Gaussian Naive Bayes (NB), and XGBoost. Each model was tuned by running 3-fold cross-validation over a set of 50 candidate

hyperparameter combinations, totalling 150 fits per model. Among the models, Random Forest and XGBoost achieved the highest cross-validation scores of 0.7994, suggesting that these models are the most suitable for the given data set. The Decision Tree and KNN models also performed reasonably well, while the Gaussian Naive Bayes model had the lowest performance with a cross-validation score of 0.6391. These results indicate that more complex models like Random Forest and XGBoost are better suited for this specific prediction task.

5.7 Model Evaluation

```
Model Comparison:
Decision Tree: Best CV Score = 0.7262
Random Forest: Best CV Score = 0.7994
KNN: Best CV Score = 0.7308
Gaussian NB: Best CV Score = 0.6391
XGBoost: Best CV Score = 0.7944

Best Model: Random Forest
Best Model CV Score: 0.7994
Classification Report
              precision    recall  f1-score   support

           0       0.78      0.79      0.78      3566
           1       0.81      0.84      0.82      5799
           2       0.84      0.82      0.83     10635

    accuracy                           0.82     20000
   macro avg       0.81      0.81      0.81     20000
weighted avg       0.82      0.82      0.82     20000
```

Figure 5.7.1 Best model

The figure above shows the Random Forest model was identified as the best model based on the cross-validation score, achieving a CV score of 0.7994. The classification report further validates the model's effectiveness, with an overall accuracy of 0.82. The model performs consistently across all classes with F1-scores ranging from 0.78 to 0.83, indicating a good balance between precision and recall. This makes the Random Forest model a robust choice for the classification task at hand.

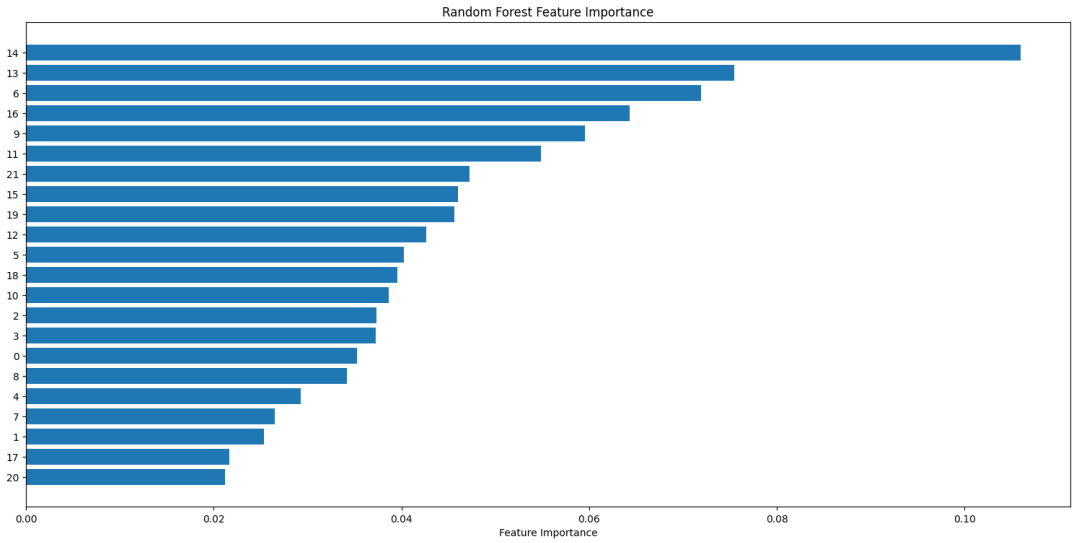


Figure 5.7.2 Feature Importance

The figure above shows the RandomForest feature importance the most significant is column 14.

Chapter 6 System Evaluation and Discussion

6.1 Interface Design

Creating an interface for your credit scoring system using Python Flask is essential to enable seamless interaction between users and the model. A user-friendly interface allows individuals or businesses to input relevant financial data and instantly receive a credit score prediction, making the system accessible to those without technical expertise. Flask provides a lightweight and flexible web framework to build this interface, allowing users to interact with your credit scoring model in real-time through a web browser. This not only enhances the usability of the model but also enables broader deployment and scalability, making it a practical tool for decision-making in real-world financial applications.

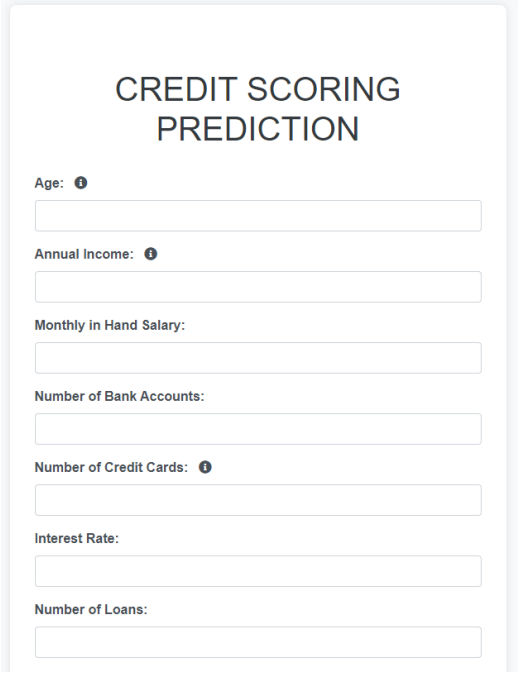


Figure 6.1.1 Interface

The Figure above shows this is the interface for credit scoring predictions to allow users to input their personal financial information and the text.csv to predict their credit score by RandomForest model and sentiment analysis.

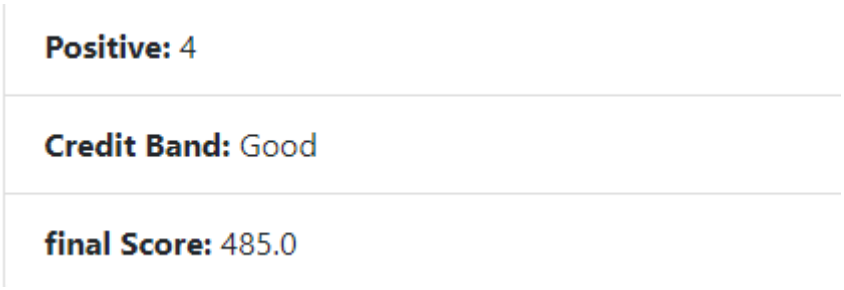


Figure 6.1.2 result

The figure above shows this is the result of the user after they input all the necessary personal financial information, the backend will run the RandomForest and Finbert to calculate the final credit score and credit band based on the user input.

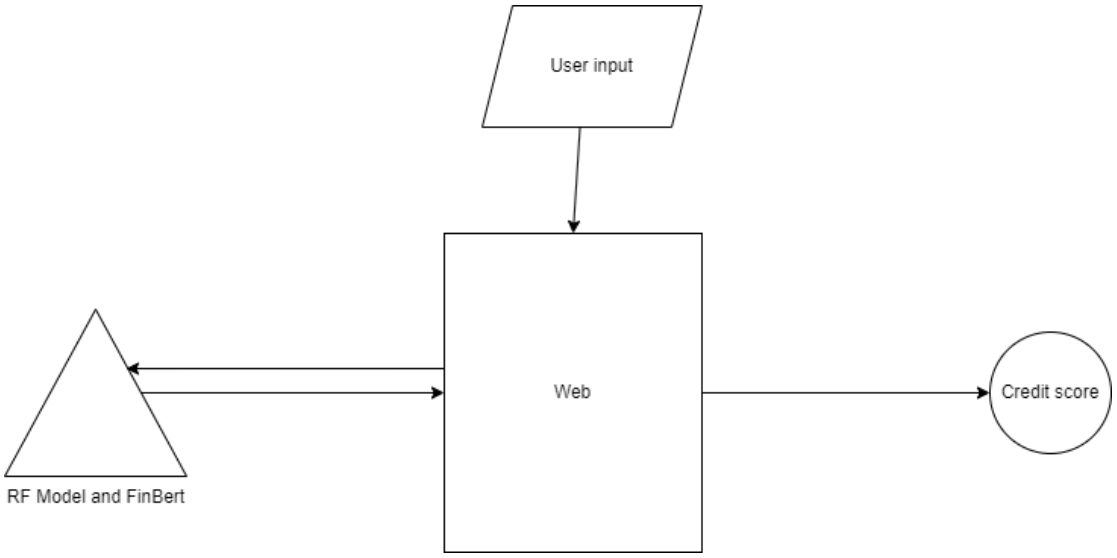


Figure 6.1.3 Interface logic

The figure illustrates a flow for the system that integrates a web interface with machine learning models, specifically a Random Forest (RF) model and FinBERT. Users interact with the system by providing input data through the web interface. This input is then processed by the RF model to handles traditional numerical and categorical data, and FinBERT used for sentiment analysis or text data processing. These two models will combined results and after the calculation will be display to the web interface, which is a credit score and credit band for the user.

6.2 Maths

```python
def calculate_credit_score(probability_of_default, min_score=300, max_score=850):
    return min_score + (max_score - min_score) * (1 - probability_of_default)
```

```python
def main(df):
    prediction, prob = predict_credit_score(df)
    x = calculate_credit_score((prob.tolist()[0][2]/2) + prob.tolist()[0][1])
    score_bands = [assign_score_band(x)]
    return score_bands[0], x
```

Figure 6.2.1 Calculation of credit score

The calculate_credit_score function computes a credit score based on the probability of default, scaling it within a range from 300 to 850. The formula adjusts the minimum score upward depending on how low the probability of default is, meaning a lower probability leads to a higher credit score.

In the main function, the process begins by using the predict_credit_score function to obtain both a prediction and the corresponding probabilities for different risk categories. The credit score is then calculated using a weighted sum of these probabilities, specifically taking half of the probability from one category (prob.tolist()[0][2]) and adding it to the full probability from another (prob.tolist()[0][1]). This combined value is passed to calculate_credit_score to generate a numerical credit score. This score is then categorized into a credit band. By the assign_score_band function, and the function returns both the descriptive score band and the exact credit score. This approach ensures that the output is not only a numerical score but also an easy-to-understand classification, making the results more accessible to users.

6.3 System Testing and Performance Metrics

A performance assessment and reliability analysis of this credit scoring application requires the use of a specialized testing dataset. This dataset was developed to replicate actual use cases allowing us to see how well the application can derive credit scores based on the input data. Credits scores are computed by the application but owing to the nature of the information being processed, the application cannot produce accuracy measures for the credit score as there are a multitude of external factors at play. Factors such as pictures interpretation and grade assessment will have given this performance outcome a subjective angle which is why the performance outcome will not be absolute. Because of their sensitive nature, both financial data and personal data cannot be precisely sourced, thus, a target of one hundred percent will always be unachievable. Instead, the application consistently though, reliably, provide usable credit score.

6.4 Testing Setup and Result

By selecting a data from testing dataset, to test the credit score how polarity score affects it the data will run on three different text.csv with positive, negative, and neutral.

```
'Age': 24,
'Occupation': 'Doctor',
'Annual_Income': 114838.41,
'Monthly_Inhand_Salary': 9843.86,
'Num_Bank_Accounts': 2,
'Num_Credit_Card': 5,
'Interest_Rate': 7,
'Num_of_Loan': 3,
'Type_of_Loan': 'Auto Loan, Debt Consolidation Loan, Not Specified',
'Delay_from_due_date': 11,
'Num_of_Delayed_Payment': 9,
'Changed_Credit_Limit': 7.1,
'Num_Credit_Inquiries': 8,
'Credit_Mix': 'Good',
'Outstanding_Debt': 1377.74,
'Credit_Utilization_Ratio': 29.17,
'Credit_History_Age': '22 Years and 1 Months',
'Payment_of_Min_Amount': 'No',
'Total_EMI_per_month': 226.89,
'Amount_invested_monthly': 916.95,
'Payment_Behaviour': 'Low_spent_Medium_value_payments',
'Monthly_Balance': 120.54
```

Figure 6.4.1 sample data 1

Most Positive	Most Negative	Most Neutral	Mix
550.8	511.2	531.0	531.6

Table 6.4.1 result of sample data 1

```
'Age': 39,
'Occupation': 'Manager',
'Annual_Income': 8701.54,
'Monthly_Inhand_Salary': 519.12,
'Num_Bank_Accounts': 6,
'Num_Credit_Card': 5,
'Interest_Rate': 32,
'Num_of_Loan': 7,
'Type_of_Loan': ''Auto Loan', 'Credit-Builder Loan', 'Debt
'Delay_from_due_date': 23,
'Num_of_Delayed_Payment': 6,
'Changed_Credit_Limit': 7.1,
'Num_Credit_Inquiries': 6,
'Credit_Mix': 'Standard',
'Outstanding_Debt': 2602.69,
'Credit_Utilization_Ratio': 28.57,
'Credit_History_Age': '9 Years and 4 Months',
'Payment_of_Min_Amount': 'Yes',
'Total_EMI_per_month': 36.54,
'Amount_invested_monthly': 52.93,
'Payment_Behaviour': 'Low_spent_Medium_value_payments',
'Monthly_Balance': 242.43
```

Figure 6.4.2 sample data 2

Most Positive	Most Negative	Most Neutral	Mix
386.0	346.4	366.2	366.8

Table 6.4.2 result of sample data 2

The results above shows the affect of the polarity score, it can affect the credit score although it only consist about 20% of whole credit score but it’s significant to affect the credit score.

6.4 Project Challenges

This project had several important constraints that influenced its development and execution. One of the main problems was the access to sensitive information like personal and financial data. This type of information is essential to perform credit scoring accurately but is usually inaccessible due to privacy issues. Moreover, there was also the issue of time, regarding working with vast text datasets especially for sentiment analysis since these techniques were data and resource intensive. The activity of modelling itself was an area where time was spent as the models required fine-tuning and validation to say the least to produce meaningful insights. In addition, some of the issues regarding data collection for Twitter's real-time sentiment analysis using APIs, were expensive since these tweets would be push notifications, and therefore contributed to an inability to analyse the live sentiments. These challenges illustrate the difficulties in merging the old school credit scoring with the emerging techniques using data.

6.5 Objectives Evaluation

The project has managed to achieve the goals and accomplish a significant step forward in improving classical credit scoring with sentiment analysis. The project properly incorporates the Natural Language Processing (NLP) techniques to analyse text data whereby polarities from user comments are determined as Positive, Negative and Neutral. This comprehensive algorithm takes a deeper dive into calculating the polarity score, which is further passed on to credit scoring, thereby utilising consuming accurately judgment of credit health. We added a web interface for sentiment prediction and visualization in real time. In this way, it becomes an interface that offers visibility into the sentiment trends and persist a much more transparent credit scoring process as well.

Chapter 7 Conclusion and Recommendation

7.1 Conclusion

In conclusion, the project tackles the problem of extending the traditional methods of credit ratings by the sentiment analysis methods integrated in one credit rating technology. The project employs natural language processing approaches and captures user sentiments which assists in overcoming the fundamental problems with existing credit scoring techniques which also helps in making the assessment of credit worthiness more multi-faceted. However, despite the challenges such as confidential nature of data, long processing time on large volumes of data, inability to rely on dynamic data fully, the project has managed to come up with its goals achieving an accurate and justifiable credit scoring system. Developed web system enables insights and predictions to be made, with the system being not only efficient and functional but friendly to use as well. In the end, this project focuses on improving in a more scientific way estimation of credit risk, which steps in supports better and fairer economics.

7.2 Recommendations

Many alterations are suggested to improve the success and reliability of the credit scoring system that has been created in this project. First, this improvement is mostly possible through the introduction of the Twitter API, since a large portion of the sentiment analysis is performed on historical data making it less relevant to the present moment. The most current users' sentiments would be captured by this system which is very important to making correct predictions of the credit scores due to the changing aspects of the social and economic conditions.

Furthermore, cause the risk characteristics of the customers to expand the data sources to other billing available for example mobile bills, social media activities on Facebook, or upon a purchase as a form of e-commerce activities. These data points would improve how credit risk is evaluated since several other determinants of the responsibility and stability of an individual would be considered. It would enable the credit scoring model to consider the social and economic factors of the people and the community enhancing the accuracy of predicting credit scores.

REFERENCES

[1] Pinakin Ariwala, “Introduction to Sentiment Analysis: Concept, Working, and Application,” Nov. 2023.

[2] A. A. Q. Aqlan, B. Manjula, and R. Lakshman Naik, “A study of sentiment analysis: Concepts, techniques, and challenges,” in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 28, Springer Science and Business Media Deutschland GmbH, 2019, pp. 147–162. doi: 10.1007/978-981-13-6459-4_16.

[3] GiniMachine, “Traditional Vs. Alternative Credit Scoring: Differences and Advantages,” Mar. 2023.

[4] Varun Mittal, “Alternative credit scoring of underbanked consumers,” Jan. 2019.

[5] Sea Bank, “Alternative Data and Credit Scoring for the Unbanked Let’s Talk Digital Series #10.”

[6] A. Sarlan, C. Nadam, and S. Basri, “Twitter sentiment analysis,” in *Conference Proceedings - 6th International Conference on Information Technology and Multimedia at UNITEN: Cultivating Creativity and Enabling Technology Through the Internet of Things, ICIMU 2014*, Institute of Electrical and Electronics Engineers Inc., 2014, pp. 212–216. doi: 10.1109/ICIMU.2014.7066632.

[7] “CalcXML.”

[8] GOPI DURGAPRASAD, “AMEX : Credit Score Model .”

[9] Z. S. V. L. Anton Markov*, “Credit scoring methods: Latest trends and points to consider,” Aug. 2022.

[10] “Natural Language Toolkit,” NLTK.

Weekly Report

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 3**
Student Name & ID: CHEW CHUN PHANG 2200944	
Supervisor: Miss Nurul Syafidah binti Jamil	
Project Title: RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS	

1. WORK DONE

- looking for suitable datasets and let supervisor checked on it.

- Established project goals and initial scope.

2. WORK TO BE DONE

-perform data preprocessing

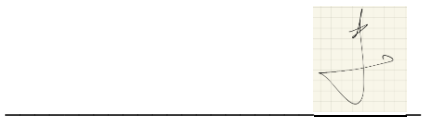
-evaluation project goals and scope

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

I have been working on understand data and modelling classifier

Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.:6**
Student Name & ID: CHEW CHUN PHANG 2200944	
Supervisor: Miss Nurul Syafidah binti Jamil	
Project Title: RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS	

1. WORK DONE

-performed data preprocessing for datasets

-choose modelling classifier

2. WORK TO BE DONE

- Tuning parameter

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

-the modelling consume a lot of time

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.:9**
Student Name & ID: CHEW CHUN PHANG 2200944	
Supervisor: Miss Nurul Syafidah binti Jamil	
Project Title: RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS	

1. WORK DONE

-report draft

- find the best parameter for modelling

-design web interface

2. WORK TO BE DONE

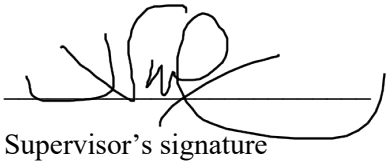
-finalize the report

-evaluation model

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

-enhance the report



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.:11**
Student Name & ID: CHEW CHUN PHANG 2200944	
Supervisor: Miss Nurul Syafidah binti Jamil	
Project Title: RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS	

1. WORK DONE

-Finalize the report

-modelling

-evaluation interface

2. WORK TO BE DONE

- application testing

3. PROBLEMS ENCOUNTERED

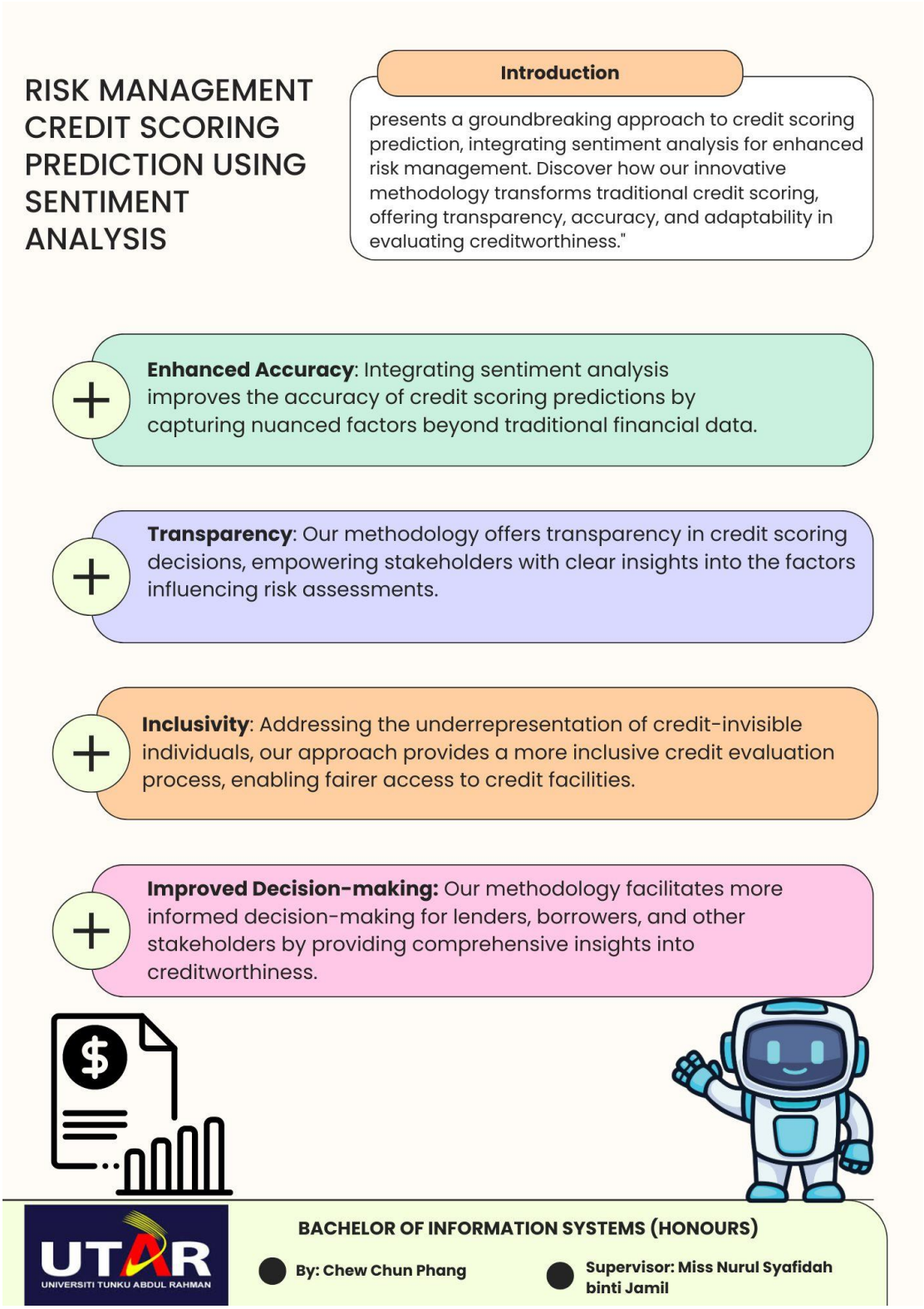
4. SELF EVALUATION OF THE PROGRESS

-check the report to ensure every part is included in report

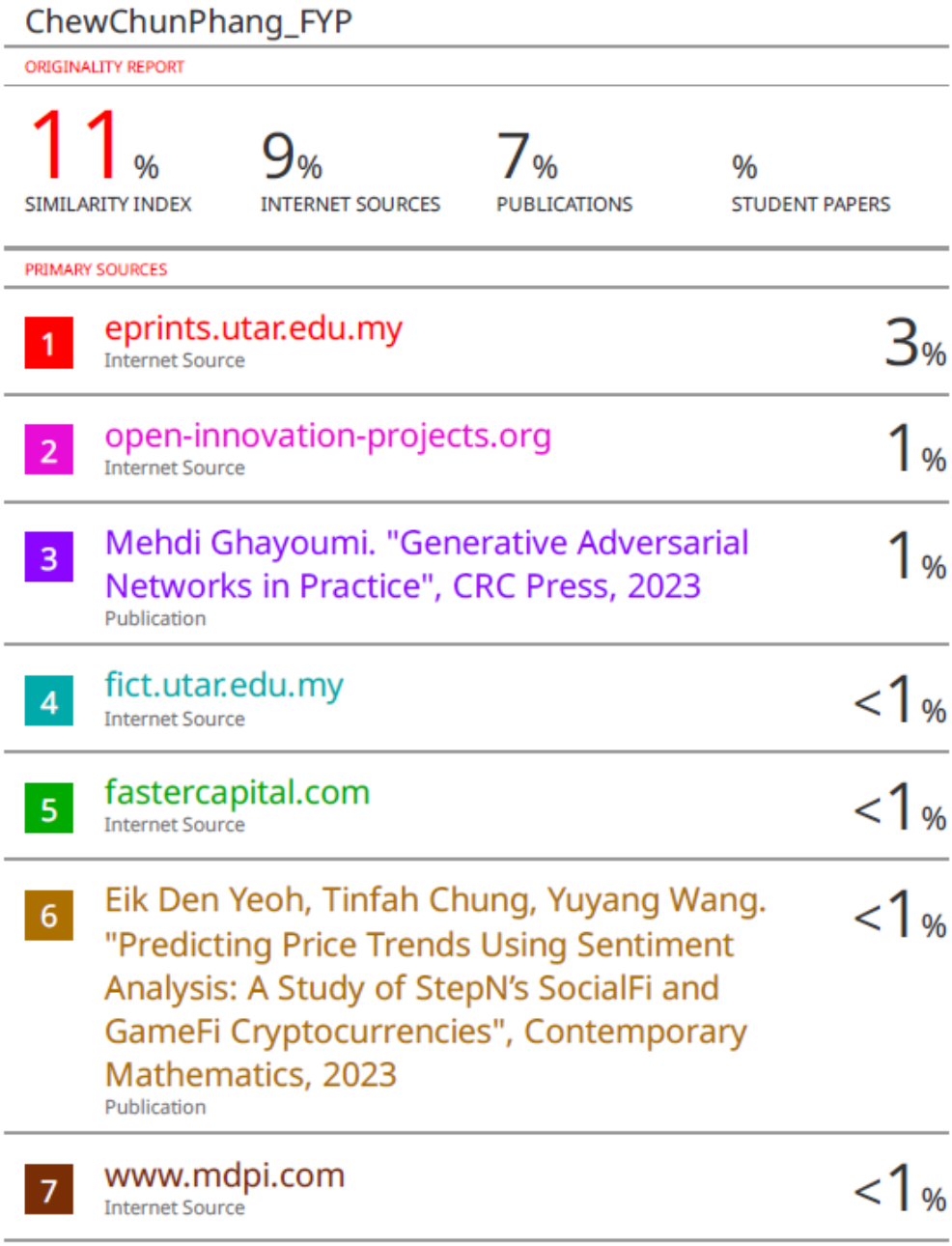
Supervisor's signature

Student's signature

POSTER


RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS
Introduction
presents a groundbreaking approach to credit scoring prediction, integrating sentiment analysis for enhanced risk management. Discover how our innovative methodology transforms traditional credit scoring, offering transparency, accuracy, and adaptability in evaluating creditworthiness."
Enhanced Accuracy: Integrating sentiment analysis improves the accuracy of credit scoring predictions by capturing nuanced factors beyond traditional financial data.
Transparency: Our methodology offers transparency in credit scoring decisions, empowering stakeholders with clear insights into the factors influencing risk assessments.
Inclusivity: Addressing the underrepresentation of credit-invisible individuals, our approach provides a more inclusive credit evaluation process, enabling fairer access to credit facilities.
Improved Decision-making: Our methodology facilitates more informed decision-making for lenders, borrowers, and other stakeholders by providing comprehensive insights into creditworthiness.
UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
BACHELOR OF INFORMATION SYSTEMS (HONOURS)
By: Chew Chun Phang
Supervisor: Miss Nurul Syafidah binti Jamil

PLAGIARISM CHECK RESULT



ChewChunPhang_FYP

ORIGINALITY REPORT

11%	9%	7%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	3%
2	open-innovation-projects.org Internet Source	1%
3	Mehdi Ghayoumi. "Generative Adversarial Networks in Practice", CRC Press, 2023 Publication	1%
4	fict.utar.edu.my Internet Source	<1%
5	fastercapital.com Internet Source	<1%
6	Eik Den Yeoh, Tinfah Chung, Yuyang Wang. "Predicting Price Trends Using Sentiment Analysis: A Study of StepN's SocialFi and GameFi Cryptocurrencies", Contemporary Mathematics, 2023 Publication	<1%
7	www.mdpi.com Internet Source	<1%

8	as-books.com Internet Source	<1%
9	Anton Markov, Zinaida Seleznyova, Victor Lapshin. "Credit Scoring Methods: Latest Trends and Points to Consider", The Journal of Finance and Data Science, 2022 Publication	<1%
10	www.coursehero.com Internet Source	<1%
11	Parikshit N. Mahalle, Namrata N. Wasatkar, Gitanjali R. Shinde. "Data-Centric Artificial Intelligence for Multidisciplinary Applications", CRC Press, 2024 Publication	<1%
12	Chinmay Chakraborty, Manisha Guduri, K. Shyamala, B. Sandhya. "Multifaceted Approaches for Data Acquisition Processing and Communication", CRC Press, 2024 Publication	<1%
13	hdl.handle.net Internet Source	<1%
14	360digitmg.com Internet Source	<1%
15	ldf.fi Internet Source	<1%

16	"Advances in Intelligent Computing Techniques and Applications", Springer Science and Business Media LLC, 2024 Publication	<1%
17	123dok.com Internet Source	<1%
18	Ali Louati, Hassen Louati, Meshal Alharbi, Elham Kariri, Turki Khawaji, Yasser Almubaddil, Sultan Aldwsary. "Machine Learning and Artificial Intelligence for a Sustainable Tourism: A Case Study on Saudi Arabia", Information, 2024 Publication	<1%
19	www.blogarama.com Internet Source	<1%
20	www.techtarget.com Internet Source	<1%
21	Urvashi Arora, Maninderjeet Singh, Siddharaj Dabade, Alamgir Karim. "Analyzing the efficacy of different machine learning models for property prediction of solid polymer electrolytes", American Chemical Society (ACS), 2024 Publication	<1%
22	www.ijirset.com Internet Source	<1%

23	www.ijraset.com Internet Source	<1%
24	link.springer.com Internet Source	<1%
25	assets-eu.researchsquare.com Internet Source	<1%
26	www.cdata.com Internet Source	<1%
27	Sanjeev J. Wagh, Manisha S. Bhende, Anuradha D. Thakare. "Fundamentals of Data Science", CRC Press, 2021 Publication	<1%
28	patents.justia.com Internet Source	<1%
29	res.mdpi.com Internet Source	<1%
30	www.kluniversity.in Internet Source	<1%
31	ijsrst.com Internet Source	<1%
32	www.lesswrong.com Internet Source	<1%
33	1login.easychair.org Internet Source	<1%

34	Arvind Dagur, Dhirendra Kumar Shukla, Nazarov Fayzullo Makhmadiyarovich, Akhatov Akmal Rustamovich, Jabborov Jamol Sindorovich. "Artificial Intelligence and Information Technologies", CRC Press, 2024 Publication	<1 %
35	listens.online Internet Source	<1 %
36	Debabrata Samanta, SK Hafizul Islam, Naveen Chilamkurti, Mohammad Hammoudeh. "Data Analytics, Computational Statistics, and Operations Research for Engineers - Methodologies and Applications", CRC Press, 2022 Publication	<1 %
37	Jan Žižka, František Dařena, Arnošt Svoboda. "Text Mining with Machine Learning - Principles and Techniques", CRC Press, 2019 Publication	<1 %
38	Syed Ali Hussain, P N S B S V Prasad V, Swikriti Khadke, Pragya Gupta, Pradyut Kumar Sanki. "Innovative Web Application Revolutionizing Disease Detection, Empowering Users and Ensuring Accurate Diagnosis", Journal of Electronic Materials, 2024 Publication	<1 %

39	ebin.pub Internet Source	<1 %
40	onlineresource.ucsy.edu.mm Internet Source	<1 %
41	readthedocs.org Internet Source	<1 %
42	strathprints.strath.ac.uk Internet Source	<1 %
43	study.utar.edu.my Internet Source	<1 %
44	www.bitcoininsider.org Internet Source	<1 %
45	www.ijritcc.org Internet Source	<1 %
46	Sufyan bin Uzayr. "Conquering JavaScript - The Practical Handbook", CRC Press, 2023 Publication	<1 %
47	R. Lakshmana Kumar, R. Indrakumari, B. Balamurugan, Achyut Shankar. "Exploratory Data Analytics for Healthcare", CRC Press, 2021 Publication	<1 %

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Chew Chun Phang
ID Number(s)	22ACB00944
Programme / Course	Digital Economy Technology
Title of Final Year Project	**RISK MANAGEMENT CREDIT SCORING PREDICTION USING SENTIMENT ANALYSIS**

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 11 % **Similarity by source** Internet Sources: 9 % Publications: 7 % Student Papers: 0 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

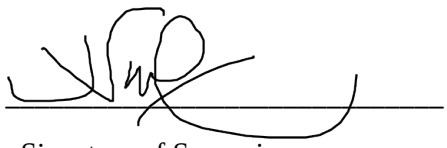
(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.



Signature of Supervisor

Name: NURUL SYAFIDAH JAMIL

Date: 11/9/2024

Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	22ACB00944
Student Name	Chew Chun Phang
Supervisor Name	Cik Nurul Syafidah Binti Jamil

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
NA	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

_____

(Signature of Student)
Date: 3/9/2024