

# **Predictive Modeling in Credit Risk Assessment**

By

Hee Tuck Hoe

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

**BACHELOR OF COMPUTER SCIENCE (HONOURS)**

Faculty of Information and Communication Technology  
(Kampar Campus)

JAN 2023

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

# REPORT STATUS DECLARATION FORM

Title: Predictive Modeling in Credit Risk Assessment  
\_\_\_\_\_  
\_\_\_\_\_

Academic Session: JAN 2023

I HEE TUCK HOE

(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:

No.181

Titi Satu, Jalan Kenas

33000, Kuala Kangsar, Perak

Date: 28/4/2023

Lim Jia Qi

Supervisor's name

Date: 28/04/2023

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

**FACULTY/INSTITUTE\* OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 28/04/2023

**SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS**

It is hereby certified that Hee Tuck Hoe (ID No: 19ACB03841) has completed this final year project/ dissertation/ thesis\* entitled “Predictive Modeling in Credit Risk Assessment” under the supervision of Dr. Lim Jia Qi (Supervisor) from the Department of Computer Science, Faculty/Institute\* of Information and Communication Technology, and Dr. Tong Dong Ling (Co-Supervisor)\* from the Department of Computer Science, Faculty/Institute\* 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,



Hee Tuck Hoe

# DECLARATION OF ORIGINALITY

I declare that this report entitled “**Predictive Modeling in Credit Risk Assessment**” 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 : Hee Tuck Hoe

Date : 28/4/2023

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

# ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, who has given me this bright opportunity to engage in this project. He has given me a lot of guidance in order to complete this project. When I was facing problems in this project, the advice from him always assists me in overcoming the problems. Again, a million thanks to my supervisor.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

# ABSTRACT

In the financial sector, particularly the banking sector, credit risk assessment is crucial. By keeping credit risk exposure within reasonable bounds, a bank's risk-adjusted rate of return can be maximized [1]. There is quite a big amount of research in credit risk, but most of the researchers only apply one model to a certain dataset and they did not develop a Graphical User Interface (GUI) for the credit risk prediction. These two will be done in this project. The main goal of this project is to develop a robust credit risk model by using several machine learning algorithms and to develop a simple GUI for the predict the borrower able to pay the loan or not. The machine learning algorithms used are logistic regression, support vector machine (SVM) and gradient boosting decision tree. These three models are then compared, and the best model is chosen to be implemented into the GUI created. The methods used to develop the model are data acquisition, data exploration and visualization, data preparation, model training which includes parameter tuning and performance evaluation. Several performance metrics will be used to evaluate the models which are accuracy, precision, recall and f1 score. The GUI is developed using Tkinter in python.

# TABLE OF CONTENTS

## Contents

REPORT STATUS DECLARATION FORM .....	ii
DECLARATION OF ORIGINALITY .....	iv
ACKNOWLEDGEMENTS .....	v
ABSTRACT .....	vi
TABLE OF CONTENTS .....	vii
LIST OF FIGURES .....	ix
LIST OF TABLES .....	xi
Chapter 1: Introduction .....	1
1.1 Problem Statement and Motivation .....	1
1.2 Project Scope .....	2
1.3 Project Objective .....	3
1.4 Contributions .....	3
1.5 Report Organization.....	4
Chapter 2: Literature Review .....	5
2.1 Review of the Technologies.....	5
2.1.1 Hardware .....	5
2.1.2 Software.....	5
2.1.3 Programming Language .....	9
2.2 Review of the Existing Systems/Applications .....	10
2.2.1 Credit Risk Assessment based on Gradient Boosting DecisionTree .....	10
2.2.2 Consumer credit risk modelling using machine learning algorithms: a comparative approach.....	12
Chapter 3: System Design.....	14
3.1 Download data.....	14
3.2 Data Exploration (EDA)/Visualisation .....	15
3.3 Data Preprocessing .....	16
3.4 Model training .....	17
3.4.1 Logistic Regression.....	17
3.4.2 Support Vector Machine (SVM).....	18
3.4.3 Gradient Boosting Decision Tree .....	19
3.5 Hyperparameter Tuning .....	20
3.5.1 Support Vector Machine (SVM).....	20
3.5.2 Gradient Boosting Decision Tree (GDBT).....	21

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

3.6 Evaluation .....	22
3.7 Develop simple Graphic User Interface (GUI).....	23
Chapter 4: Experiment/Simulation.....	24
4.1 Software Setup .....	24
4.1.1 Anaconda and Jupyter Notebook Installation .....	24
4.2 System Implementation.....	25
4.2.1 Data Exploration .....	25
4.2.2 Data Visualisation .....	28
4.2.3 Feature Engineering.....	30
4.2.4 Model Training.....	31
4.2.5 Hyperparameter Tuning .....	35
4.2.6 Performance Evaluation .....	37
4.2.7 Develop simple Graphic User Interface (GUI).....	38
4.3 Implementation Issues and Challenges .....	40
4.4 Concluding Remark.....	41
Chapter 5: System Evaluation and Discussion.....	42
5.1 System Testing and Performance Evaluation of Models .....	42
5.1.1 Accuracy, Precision, Recall &F1 score.....	42
5.1.2 Classification report & Confusion matrix.....	44
5.1.1 AUC-ROC curve .....	48
5.2 Testing Setup and Result .....	51
5.3 Project Challenges .....	54
5.3 Objectives Evaluation .....	55
5.3 Concluding Remark.....	56
Chapter 6: Conclusion and Recommendation.....	57
6.1 Conclusion.....	57
6.2 Recommendation .....	58
REFERENCE.....	57
APPENDIX.....	59
WEEKLY REPORT .....	74
POSTER.....	80
PLAGIARISM CHECK RESULT.....	81
REPORT CHECKLIST .....	83



# LIST OF FIGURES

Figure 2. 1: Icon of anaconda in the centre, jupyter notebook at left and spyder at right....	5
Figure 2. 2: Icon of NumPy .....	6
Figure 2. 3: Icon of Scikit-learn .....	6
Figure 2. 4: Icon of Pandas .....	7
Figure 2. 5: Icon of Matplotlib.....	8
Figure 2. 6: Icon of Joblib .....	8
Figure 2. 7: Icon of Python programming language .....	9
Figure 2. 8: Evaluation of Predictions.....	10
Figure 2. 9: Evaluation of Predictions.....	12
Figure 3. 1: Research Workflow .....	14
Figure 3. 2: Training of gradient boosting decision tree. ....	17
Figure 3. 3: Finding the hyper-plane that differentiates the two classes.....	18
Figure 3. 4: Hyper-plane in original input space looks like a circle .....	19
Figure 3. 5: Training of gradient boosting decision tree. ....	19
Figure 4. 1: Code to open jupyter notebook.....	24
Figure 4. 2: Choose Python 3 .....	25
Figure 4. 3: Result of converting numeric value into categoric. ....	27
Figure 4. 4: Result of boxplots.....	28
Figure 4. 5: Result of Correlation Matrix .....	28
Figure 4. 6: Result of contingency table .....	29
Figure 4. 7: Result of encoding .....	30
Figure 4. 8: Result of normalization .....	30
Figure 4. 9: Result of sampling data with SMOTE approach .....	31
Figure 4. 10: Split the dataset into training and testing data.....	31
Figure 4. 11: Train the data with Logistic Regression algorithm .....	31
Figure 4. 12: Split the dataset into training and testing data.....	32
Figure 4. 13: Train the data with Linear Kernel SVM algorithm .....	32
Figure 4. 14: Split the dataset into training and testing data.....	33
Figure 4. 15: Train the data with RBF Kernel SVM algorithm .....	33
Figure 4. 16: Split the dataset into training and testing data.....	34
Figure 4. 17: Train the data with GBDT algorithm .....	34
Figure 4. 18: Hyperparameter tuning the Linear Kernel SVM .....	35
Figure 4. 19: Hyperparameter tuning the RBF Kernel SVM.....	36
Figure 4. 20: Hyperparameter tuning the GBDT model .....	37
Figure 4. 21: Coding of Accuracy, Precision, Recall & F1 Score .....	38
Figure 4. 22: Coding of Confusion Matrix & Classification Report .....	38
Figure 4. 23: Coding of AUC-ROC curve .....	38
Figure 4. 24: GUI for credit risk prediction .....	39
Figure 4. 25: Example result of prediction.....	39
Figure 4. 26: Prediction detail saved in text file .....	40

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

Figure 5. 1: Test Accuracy, Precision, Recall & F1 score of Logistic Regression model..	42
Figure 5. 2: Test Accuracy, Precision, Recall & F1 score of Linear kernel SVM model...	42
Figure 5. 3: Test Accuracy, Precision, Recall & F1 score of Linear kernel SVM model after hyperparameter tuning .....	42
Figure 5. 4: Test Accuracy, Precision, Recall & F1 score of RBF kernel SVM model.....	42
Figure 5. 5: Test Accuracy, Precision, Recall & F1 score of RBF kernel SVM model after hyperparameter tuning .....	42
Figure 5. 6: Test Accuracy, Precision, Recall & F1 score of GBDT model .....	43
Figure 5. 7: Test Accuracy, Precision, Recall & F1 score of GBDT model after hyperparameter tuning .....	43
Figure 5. 8: Classification report & Confusion matrix of Logistic Regression model .....	44
Figure 5. 9: Classification report & Confusion matrix of Linear kernel SVM model.....	44
Figure 5. 10: Classification report & Confusion matrix of Linear kernel SVM model after hyperparameter tuning .....	45
Figure 5. 11: Classification report & Confusion matrix of RBF kernel SVM model.....	45
Figure 5. 12: Classification report & Confusion matrix of RBF kernel SVM model after hyperparameter tuning .....	46
Figure 5. 13: Classification report & Confusion matrix of Gradient Boosting Decision Tree model .....	46
Figure 5. 14: Classification report & Confusion matrix of Gradient Boosting Decision Tree model after hyperparameter tuning .....	47
Figure 5. 15: AUC-ROC curve of Logistic Regression model .....	48
Figure 5. 16: AUC-ROC curve of Linear kernel SVM model.....	48
Figure 5. 17: AUC-ROC curve of Linear kernel SVM model after hyperparameter tuning .....	49
Figure 5. 18: AUC-ROC curve of RBF kernel SVM model.....	49
Figure 5. 19: AUC-ROC curve of RBF kernel SVM model after hyperparameter tuning .	50
Figure 5. 20: AUC-ROC curve of Gradient Boosting Decision Tree model .....	50
Figure 5. 21: AUC-ROC curve of Gradient Boosting Decision Tree model after hyperparameter tuning .....	51
Figure 5. 22: GUI created for credit risk prediction.....	52
Figure 5. 23: System will show error message if there is blank field found.....	52
Figure 5. 24: System will display the result at the bottom right .....	53
Figure 5. 25: Test 1 .....	54

# LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 2.1	Specifications of laptop	5
Table 4.1	Name, type and description of the attributes and whether there is missing value or not.	25-27
Table 5.1	Result of Prediction	53
Table 5.2	Summary of Evaluation	56

## **Chapter 1: Introduction**

Credit risk assessment is the process of determining how likely it is that a borrower would not pay back a loan or other credit obligation within a particular timeframe. It entails examining several aspects of the borrower's financial condition to assess the risk involved in making a loan to them. The goal of credit risk assessment is to assist lenders in managing their risks and choosing wisely how much money to lend. The study of credit risk assessment is centered on the numerous approaches and instruments used by lenders to assess potential borrowers' creditworthiness and control their risks. This includes using credit scoring models, machine learning algorithms, and other techniques to anticipate the possibility of default, as well as quantitative and qualitative analyses of aspects like credit history, income, debt levels, and payment habits. Understanding how lenders evaluate credit risk and how this influences lending decisions and access to credit for people and enterprises is the goal of the study of credit risk assessment.

### **1.1 Problem Statement and Motivation**

Credit risk assessments have existed for a long time. The development of more formal methods of credit risk assessment was facilitated by the expansion of the banking sector in the 19th century. The Dun & Bradstreet credit rating system, which was first used in the United States in 1859, is one famous example. With the use of this system, lenders may more easily determine the creditworthiness of prospective borrowers by giving firms a credit rating based on their payment history, financial stability, and other variables. Credit risk modelling has been advancing quickly in recent years to become a critical component of financial organizations' risk management systems and has been a research hotspot among machine learning practitioners. However, the model they applied for the credit risk assessment is very specific to a certain dataset. When the model is used for other datasets, the result may not be comparable to the previous data, which indicates the low generalizability of the models. In other words, the models they applied are not robust enough and are overfitting. Thus, for this project, a few robust models for credit risk assessment are trained. The models that were developed are logistic regression model, gradient boosting decision tree model and support vector machine (SVM) model.

Another problem statement is the project has focused solely on developing the model without creating a graphical user interface (GUI) or implementing the model into the Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

GUI. The absence of a GUI and integration of the model into it poses limitations and restricts the practical usability of the credit risk assessment system. Without a user-friendly interface, stakeholders, such as loan officers or risk managers, are unable to interact with the model and utilize its predictive capabilities effectively. Additionally, the lack of a GUI prevents easy deployment of the model in real-world scenarios. The problem at hand requires the development of a GUI that seamlessly incorporates the credit risk assessment model. The GUI should provide an intuitive interface for users to input customer information, trigger the model's predictions, and obtain risk assessments promptly. The implementation of the model within the GUI should enable users to obtain accurate predictions and risk scores in a user-friendly manner. Addressing this problem will enhance the usability and practicality of the credit risk assessment system, empowering stakeholders to make informed lending decisions efficiently. By bridging the gap between the developed model and its application through a GUI, the project aims to provide an end-to-end solution that can be readily adopted in real-world credit risk assessment scenarios.

## **1.2 Project Scope**

The scope of the project includes supervised classification algorithms, data exploration, feature engineering, model training, etc. In supervised learning, algorithms learn from labelled data [2]. After analysing the data and training the models, the models adapt to the data pattern, the algorithm decides which label should be applied to the new data. For this credit risk assessment, supervised classification models are used. If the classification problem has only two possible outcomes, then it is called a Binary Classifier. If a classification problem has more than two outcomes, then it is called a Multi-class Classifier. The credit risk assessment problem falls under the Binary classification.

Data exploration, feature engineering, model training, performance evaluation and parameter tuning are the crucial steps in the machine learning pipeline which will be used to complete this project. All these steps and methodologies used will be further discussed in Chapter 3.

At the end of the project, a graphical user interface (GUI) will be developed to test the model with real-time data. This GUI will be created using Tkinter modules in python language.

There are some fields of study that are not going to be covered in this project such as deep learning. A kind of machine learning called "deep learning" is based on how the human brain is organised and functions [3]. In deep learning, neural networks are utilised to perform intricate calculations on enormous amounts of data. A neural network is structured like the human brain using synthetic neurons, also referred to as nodes. These nodes are stacked on top of three layers: the input layer, hidden layer, and output layer. Deep learning comes in many forms. For instance, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep Belief Networks (DBNs), and so forth. This will become a limitation of this study.

### **1.3 Project Objective**

There are primarily 2 objectives of this study:

- To develop a robust binary classifier for credit risk assessment. Logistic Regression, gradient boosting decision tree and Support Vector Machine will be used for the credit risk assessment. These three models will be compared and assessed at the end of the modelling process using a variety of performance metrics such as the confusion matrix, AUC-ROC curve, F1 score, accuracy, and so on.
- To develop a graphical user interface (GUI) for the credit risk prediction using Tkinter library in python. After comparing the three models, the model with the best performance will be chosen and deployed into the GUI to predict the credit risk using real data.

### **1.4 Contributions**

The contributions of this project are outlined as below:

- If a robust classifier for credit risk modelling is developed, the benefits are fast online loan application validation. Nowadays, some loan firms can approve loan applications within an hour existing six-month old customer / borrower. This is most probably because they have their own credit risk model. The credit risk model helps them to achieve validation faster compared to the traditional method. The traditional method involves verification and validation process which are time consuming. The loan firm only needs to input the information of borrower into the model, the model will compute the output more specifically the probability score. Although the human still has to do

## Chapter 1 Introduction

some analysis, the analysis is less time consuming because most of the heavy load has been done by the model already. The human only needs to interpret the output of the model instead of screening through the whole information of each applicant.

- The GUI developed helps to predict the credit risk using real data. The loan firms input the information data of the borrower into the GUI, it will help them to predict whether they accept or reject the application of the borrowers.
- Dealing with the imbalanced data using Synthetic Minority Oversampling Technique (SMOTE) approach. This SMOTE approach will increase the sample of default class so that the data will be more balanced. When there is more sample of default class, the result will be more accurate in detecting the default cases. In other words, the SMOTE approach will help to boost the accuracy of the model.

### **1.5 Report Organization**

This project is organized into 4 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology and Approach, Chapter 4 System Design. The first chapter is the introduction of this project which includes problem statement, project background and motivation, project scope, project objectives, project contribution, and report organization. The second chapter is the literature review carried out on several existing credit risk assessments to evaluate the result of each of the research. In this chapter, the technologies such as the hardware platform, database, algorithm, programming language, etc. used in the project are also reviewed. The third chapter will discuss the system design of the project. The methodology and the workflow of this project will be discussed in detail. Chapter 4 will discuss about the implementation and how the project is carried out. Chapter 5 shows the result of the three models and the GUI created. Chapter 6 wraps up the whole project with a conclusion and recommendation.

## Chapter 2: Literature Review

### 2.1 Review of the Technologies

#### 2.1.1 Hardware

Description	Specifications
Model	Acer Aspire E5-476G
Processor	Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
Operating System	Windows 10
Graphic	NVIDIA GeForce MX150 with 2GB VRAM
Memory	12GB DDR4 RAM
Storage	1000 GB HDD

Table 2. 1: Specifications of laptop

The hardware involved in this project is a laptop. The specifications of the laptop are shown in the table below.

#### 2.1.2 Software

##### 2.1.2.1 Anaconda (Jupyter Notebook)



Figure 2. 1: Icon of anaconda in the centre, jupyter notebook at left and spyder at right

Python version 3.8.8 will be used for this credit risk assessment. This will be performed in Anaconda. An open and free source platform for the Python and R programming languages is called Anaconda [4]. It offers over 1500 Python and R data science tools that are appropriate for creating deep learning and machine learning



models. Python is included in the Anaconda distribution along with a number of IDEs, such as Jupyter Notebook, Anaconda prompt, Spyder and others.

### 2.1.2.2 NumPy



Figure 2. 2: Icon of NumPy

The python libraries that will be used are NumPy, Scikit-learn, Pandas and Matplotlib. NumPy provides capabilities for handling data and numbers [5]. With the help of a large number of sophisticated mathematical functions, this well-known Python library enables you to process big multidimensional arrays and matrices. It is useful for machine learning's foundational scientific computations [6].

### 2.1.2.3 Scikit-learn



Figure 2. 3: Icon of Scikit-learn

Scikit-learn is a popular Python machine learning toolkit because of its diverse use-cases and sophisticated features [5]. The main goal of Scikit-learn is to provide efficient data analysis tools, and the library is built on top of other powerful libraries like NumPy, SciPy, and matplotlib, with support for plotly, pandas, and many others.

Scikit-learn has data classification and organising features. Regression, dimensionality reduction, clustering, data preprocessing and model selection are all included. In addition, the library includes support vector machines, gradient boosting and random forests, among other commonly used machine learning algorithms.

#### 2.1.2.4 Pandas



Figure 2. 4: Icon of Pandas

A well-liked Python package for data analysis is called Pandas [6]. A two-dimensional, size-mutable tabular data format called a Pandas DataFrame has named axes and has the potential to hold heterogeneous data. The three main components of a Pandas DataFrame are the data, rows, and columns. In the real world, a Pandas DataFrame is constructed by importing the datasets from preexisting storage, such as a SQL database, a CSV file, or an Excel file. Among other things, lists, dictionaries, and lists of dictionaries can be used to create Pandas DataFrames. Dealing with the rows and columns of the dataframe is the most fundamental action that can be done on it. It is possible to delete, select, rename, etc. the columns and rows. Working with missing data, indexing and choosing data, and iterating over rows and columns are other actions that can be carried out on a dataframe. As is common knowledge, the dataset needs to be ready before training. Pandas is advantageous in this situation because it was created specifically for data preprocessing and extraction. It offers high-level data structures in addition to a wide range of data analysis tools. It contains numerous internal data gathering, merging, and filtering features. With Pandas, Python will gain a significant amount of data analysis capability, and users will be able to do complex operations without having to switch to a more specialised language [5]. Pandas comes with capabilities for time series functionality, indexing data, altering data sets, merging and combining data sets, and reading and writing data in memory data structures.

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR



Figure 2. 5: Icon of Matplotlib

Matplotlib is a very popular Python library for data visualization [6]. The Matplotlib library adds strong visualisation tools to Python, extending its capability [5]. It is a library for creating graphs and plots in two dimensions [6]. Programmers can easily create charts thanks to Python's pyplot library, which gives them the ability to customise line styles, font properties, axes formatting, etc. Line charts, bar charts, scatter plots, and histograms are among the plots that may be constructed using the library.

#### 2.1.2.6 Joblib



Figure 2. 6: Icon of Joblib

Joblib is a collection of Python pipelining tools that offers simple pipelining [7]. Joblib provides particular optimizations for numpy arrays and is designed to be quick and reliable with large data in particular. Joblib has several features, the main feature that will be used in this project is fast compressed persistence where there will be a replacement for pickle to work efficiently on Python objects containing large data using the "joblib.dump" and "joblib.load" function.

#### 2.1.2.7 Tkinter

Tkinter is the standard GUI library for Python. The combination of Python and Tkinter makes it quick and simple to develop GUI apps [8]. An effective object-oriented interface for the Tk GUI toolkit is provided by Tkinter. Tkinter offers a

number of controls, including text boxes, labels, and buttons that are used in GUI applications. Widgets are a frequent name for these controls. Tkinter now supports 15 different widget kinds, including Button, Canvas, Checkbutton, Entry, Label, etc. To arrange widgets throughout the parent widget area, specific geometry management techniques are available to all Tkinter widgets. Tkinter provides the classes pack, grid, and place as geometry managers. Before putting them in the parent widget, the pack() method divides widgets into blocks. The grid() method arranges widgets in the parent widget in a table-like fashion. Widgets are arranged using the put() method by assigning them to a certain location within the parent widget.

### 2.1.3 Programming Language



Figure 2. 7: Icon of Python programming language

Python is an interpreted, object-oriented, high-level, dynamically semantic programming language [9]. It is particularly desirable for Rapid Application Development as well as for usage as a scripting or glue language to tie existing components together due to its high-level built-in data structures, dynamic typing, and dynamic binding. Python's straightforward syntax prioritizes readability and makes it simple to learn, which lowers the cost of program maintenance. Python's support for modules and packages promotes the modularity and reuse of code in programs. For all popular platforms, the Python interpreter and the comprehensive standard library are freely distributable and available in source or binary form.

## 2.2 Review of the Existing Systems/Applications

### 2.2.1 Credit Risk Assessment based on Gradient Boosting DecisionTree

#### 2.2.1.1 Brief Overview

Tian et al. proposed a credit risk assessment model based on gradient boosting decision trees (GBDT) for predicting the default risk of loans [10]. The study used a dataset from a Chinese peer-to-peer (P2P) lending platform, which contained information on borrower characteristics, loan terms, and credit histories.

The authors applied feature engineering techniques to preprocess the data, and then used GBDT to build a predictive model for credit risk assessment. They compared the performance of their model with other popular machine learning algorithms, which are Logistic Regression, Support Vector Machine (SVM), Multilayer Perceptron (MLP), Classification And Regression Tree (CART), AdaBoost and Random Forests.

The results showed that the GBDT model outperformed the other algorithms in terms of predictive accuracy, f1 score and area under the curve (AUC) which are 90.99%, 90.37% and 0.97 respectively. The study also found that borrower characteristics, such as credit scores and income, were the most important predictors of credit risk.

	accuracy	f1 score	AUC
Logistic Regression	74.43%	74.37%	0.84
SVM	77.64%	77.94%	0.87
Decision Tree	84.68%	84.71%	0.85
MLP	84.61%	83.45%	0.93
AdaBoost	87.67%	87.37%	0.95
Random Forest	88.96%	88.45%	0.96
Gradient Boosting Decision Tree	90.99%	90.37%	0.97

Figure 2. 8: Evaluation of Predictions

Overall, the study demonstrated the effectiveness of GBDT in credit risk assessment and provided insights into the factors that contribute to default risk in P2P lending.

### **2.2.1.2 Strength**

- **Novel approach:** The study proposed a new approach to credit risk assessment using gradient boosting decision trees, which may provide better performance than traditional methods.
- **Real-world dataset:** The study used a real-world dataset from a Chinese P2P lending platform, which provides a more realistic scenario than synthetic or simulated datasets.
- **Imbalanced data:** The study used Synthetic Minority Oversampling Technique (SMOTE) method to handle the imbalanced data which might affect the accuracy of the prediction.
- **Comparison with other algorithms:** The study compared the performance of the gradient boosting decision tree model with other popular machine learning algorithms, which provides a basis for evaluating the relative performance of different methods.

### **2.2.1.3 Weakness**

- **Lack of transparency:** The study did not provide detailed information on the specific parameters and settings used in the gradient boosting decision tree model, which may make it difficult to replicate or interpret the results.
- **Overfitting:** The study did not explicitly address the potential for overfitting in the gradient boosting decision tree model, which may be a concern given the large number of features in the dataset.
- **Potential confounding variables:** The study did not account for potential confounding variables or other factors that may affect credit risk, such as macroeconomic trends or industry-specific factors.

### **2.2.1.4 Recommendation**

- **Transparency and reproducibility:** Future research could provide more detailed information on the specific parameters and settings used in the gradient boosting decision tree model, as well as provide open-source code to ensure transparency and reproducibility of the results.
- **Addressing overfitting:** Future research could explore methods to address overfitting in the gradient boosting decision tree model, such as regularization or feature selection techniques.

## Chapter 2 Literature Review

- Accounting for confounding variables: Future research could investigate the impact of potential confounding variables or other factors that may affect credit risk, such as macroeconomic trends or industry-specific factors, to provide a more comprehensive understanding of credit risk assessment.

### 2.2.2 Consumer credit risk modelling using machine learning algorithms: a comparative approach

#### 2.2.2.1 Brief Overview

In this study, Nyangena compared the performance of several machine learning algorithms for consumer credit risk modeling [11]. The algorithms evaluated included Logistic Regression, Random Forest, Gradient Boosting, Support Vector Machine and artificial neural networks (Multilayer Perceptron).

The study used a dataset of consumer loan applications from a financial institution in Kenya. The performance of each algorithm was evaluated based on its accuracy, precision, recall, f1 score and area under the receiver operating characteristic curve (AUC-ROC).

Model	Precision Score	Recall Score	F1_score	Accuracy	PR AUC
Logistic Regression	0.7778	0.0052	0.0104	0.7814	0.4006
Random Forest	0.4386	0.2058	0.2802	0.7680	0.4932
SVM	0.6373	0.0921	0.1609	0.7893	0.4643
Gradient Boosting	0.6012	0.1422	0.2300	0.7911	0.4658
MLP	0.5579	0.1369	0.2200	0.7869	0.4421

Figure 2. 9: Evaluation of Predictions

The results of the study showed that gradient boosting and artificial neural networks (MLP) outperformed the other algorithms in terms of accuracy, AUC-ROC, and sensitivity. The study concluded that machine learning algorithms can be effective in modeling consumer credit risk and that gradient boosting and artificial neural networks are particularly promising methods for this task.

#### 2.2.2.2 Strength

- The study used a real-world dataset of loan applications from a financial institution in Kenya, which increases the external validity of the study.

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

## Chapter 2 Literature Review

- The study compared multiple machine learning algorithms for credit risk modeling, which provides a comprehensive evaluation of the different methods.
- The study used multiple performance metrics, including accuracy, sensitivity, specificity, and AUC-ROC, which provides a more nuanced understanding of the algorithms' performance.

### **2.2.2.3 Weakness**

- The study only used one dataset from a single financial institution in Kenya, which may limit the generalizability of the findings to other contexts.
- The study did not provide a detailed explanation of the feature engineering process, which may affect the performance of the machine learning algorithms.
- The study did not evaluate the interpretability of the different algorithms, which may be important in practice if lenders need to explain their credit decisions to customers.

### **2.2.2.4 Recommendation**

- Replicate the study using multiple datasets from different financial institutions in different regions to increase the external validity of the findings.
- Provide a detailed explanation of the feature engineering process and conduct sensitivity analyses to test the robustness of the findings.
- Evaluate the interpretability of the different machine learning algorithms and compare them to more traditional credit risk modeling methods, such as logistic regression.



## Chapter 3: System Design

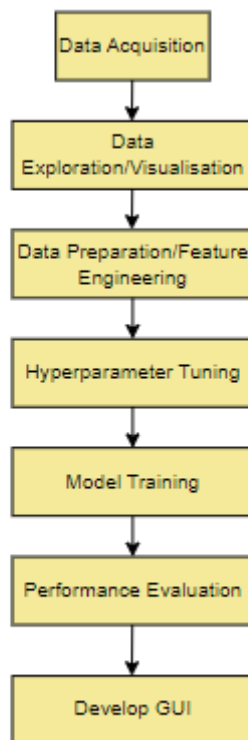


Figure 3. 1: Research Workflow

### 3.1 Download data

This is the very first step of the project. In this step, datasets for the project are collected. A data analyst should make sure that the source of data collected is trustworthy. This is because the quality of the data will affect the accuracy of the models.

For this Final Year Project, open source real dataset was used. The dataset was downloaded from an open-source repository called Kaggle. I searched the keyword to find the datasets on the Kaggle and found a lot of datasets related to credit risk predictions. After looking and exploring all the datasets, I have chosen a dataset which I think is the most suitable and reliable for this project. The file name of the dataset is “default\_of\_credit\_card\_clients”. The URL link for the dataset is “<https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset>”.

Default payments, demographic information, credit data, payment history, and bill statements for credit card users in Taiwan from April 2005 to September 2005 are all included in this dataset.

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

### **3.2 Data Exploration (EDA)/Visualisation**

The second phase is data exploration and visualisation. Exploratory data analysis which is also known as EDA is conducted in this phase with the help of python libraries such as Pandas. EDA is a type of data analysis that makes use of various graphical techniques to maximise insight into data sets, find underlying patterns, and identify abnormalities. EDA's goal is to assist analysts in comprehending data prior to drawing any conclusions. It aids in identifying data anomalies, exposing hidden patterns in the data, testing hypotheses, choosing the best features, and constructing models. Data visualisation is also done in this phase to further understand the data of the datasets. There were two types of visualization which are univariate and multivariate. Examples of univariate visualisation are histograms, line charts and boxplots. The examples of multivariate visualisation are stacked bar charts, and scatter plots.

The dataset is read with the help of Pandas library. It is loaded the data into Jupyter Notebook working environment. The dataset is explored to identify the data types of each attribute. There are two data types which are categorical and numeric. Categorical variables contain qualitative data that is restricted to a finite set of known values and does not have any meaning in calculation. While numeric variables contain quantitative data which is expressed in numbers, either continuous or discrete. During the exploration, it is found that the data of both datasets consist of both categorical and continuous features. For this dataset, the categorical features are sex, marriage, education, and previous monthly payment status. While the continuous features are amount of given credit, age, amount of bill statement in previous months, and amount of previous payment in previous months. Data is explored to check whether there were missing or noisy values. The missing or noisy values will be handled in the data preparation phase.

The visualization of data is done to further understand the datasets with the help of Scikit-learn and Matplotlib libraries. For this dataset, histogram, boxplots and correlation matrix are used to visualize the continuous features. Histogram is another form of bar charts used to display continuous categories, such as consecutive range of values for age. A box plot or a box and whisker plot is a convenient way to illustrate key descriptive statistics by showing the range of the most common data values for multiple variables. A correlation matrix is a table that shows the correlation

coefficients for various variables [12]. The correlation between all potential pairs of values in a table is shown in the matrix. It is an effective tool for compiling a sizable dataset and for locating and displaying data patterns. Contingency table is used to visualize the categorical features.

### 3.3 Data Preprocessing

Data preprocessing is the process of cleaning and transforming data so that it is ready for analysis. This phase consists of data pre-processing and feature engineering. This phase involves noise/outlier removal, handling of missing data, dummy encoding, handling of imbalanced data, dimensionality reduction, and identifying valuable attributes. There are four types of data pre-processing which are data cleaning, data integration, data reduction and data transformation.

For this dataset, after data exploration and visualization, I found that there is *no missing values* in this dataset.. All the data is stored in numeric format, which could be inappropriate for some categorical predictors. The values of numerical categorical features which are “SEX”, “EDUCATION”, “MARRIAGE”, “PAY\_0”, “PAY\_2”, “PAY\_3”, “PAY\_4”, “PAY\_5” and “PAY\_6” converted into categorical values according to the information given at the source of data. For example, the value “1” and “2” are converted to “Male” and “Female” respectively. Categorical feature encoding is done using the using the `get_dummies()` function from Pandas library. Categorical feature encoding must be done because only numerical variables are accepted by most machine learning models. Hence, we must convert these categorical variables to numbers as models like logistic regression and SVM cannot handle features stored as string. During the exploration of data, I found that the values of numerical features have a large range. Hence, the data are normalised using min-max normalization to convert the values to a common scale (What is the range?). This normalization is implemented using the `MinMaxScaler()` function from the Scikit-learn library. Besides that, the data of default class is imbalanced. Due to insufficient instances of the minority class, imbalanced classification has the drawback that a model cannot efficiently learn the decision boundary [13]. The minority class's examples can be oversampled as one approach to resolving this issue. Simple replication of samples from the minority class in the training dataset before model fitting can do this. Although it can balance the class distribution, this doesn't give the model any new data. Instead, fresh examples can be created by synthesizing the old

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

ones. The Synthetic Minority Oversampling Technique (SMOTE) is a method of data augmentation for the minority class, which is available as one of the sklearn modules.

### 3.4 Model training

In supervised learning, the basic objective of model training is to create the best model of the relationship between input attributes and a target label. For this credit risk assessment, a few algorithms which are logistic regression (LR), gradient boosting decision tree and support vector machine are implemented. All these algorithms are imported from Scikit-learn library.

#### 3.4.1 Logistic Regression

Logistic regression is a binary classification algorithm that is used when estimating a probability that a certain instance or set belongs to a specific class [14]. For example, a bank builds a model to generate the credit score of a certain client, it will model whether the credit score is “high” or “low”.

Working with logistic regression as a binary classifier, given input  $X$ , the model gives us a prediction,  $\hat{y}$ , where

$$\hat{y} = P(y = 1|X)$$

Where  $\hat{y} \in [0,1]$ . and  $y=1$  denotes high risk. The prediction,  $\hat{y}$ , can also be written as

$$\hat{y} = \sigma(w^T X + b) \text{ and}$$

where  $\sigma$  is the sigmoid function. Assume that  $z = w^T x + b$ , therefore

$\sigma(z) = 1 / (1 + e^{-z}) = e^z / (1 + e^z)$ . The plot for sigmoid function is shown in Figure 1.1.

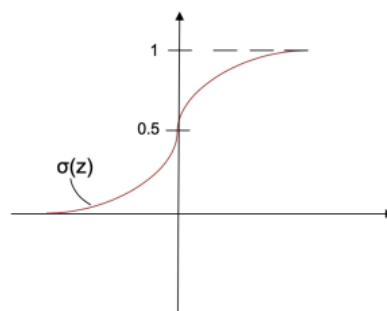


Figure 3. 2: Training of gradient boosting decision tree.

Next step is to find parameters  $w = (w^1, \dots, w^n)$  and  $b$  so that  $\hat{y}$  is close to zero when  $y = 0$  and  $\hat{y}$  is close to one when  $y = 1$ . When the model is in use for prediction and updating of  $\hat{y}$  is applied the following boundaries are used:

$$\hat{y} = 0, \text{ if } \hat{y} < 0.5$$

$$\hat{y} = 1, \text{ if } \hat{y} \geq 0.5$$

### 3.4.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the supervised machine learning methods that can be used to address classification and regression issues [15]. However, it is mostly used to address classification-related issues. The SVM approach represents each data point as a point in an  $n$ -dimensional space where  $n$  is the number of features and each feature's value denotes a specific position. The next step in classification is to locate the hyper-plane that clearly delineates the two categories.

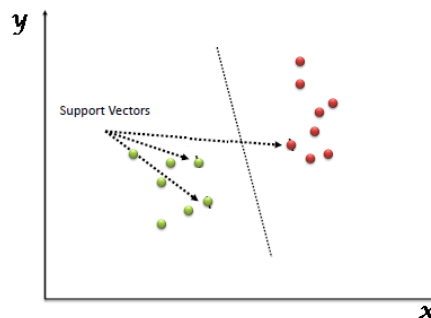


Figure 3. 3: Finding the hyper-plane that differentiates the two classes

Support vectors are the coordinates of each unique observation, to put it simply. The SVM classifier is a frontier that most successfully distinguishes between the two classes (hyper-plane/line). Even when the data is not otherwise linearly separable, SVM map data to a high-dimensional feature space so that data points can be categorised [16]. The data are transformed when a separator between the categories is found so that the separator can be depicted as a hyperplane. The category to which a new record should belong can then be determined using aspects of recent data. The SVM algorithm employs a method known as the kernel trick [16]. The SVM kernel, for instance, can be used to convert a not separable problem into a separable problem by taking a low-dimensional input space and turning it into a higher-dimensional space. Issues involving non-linear separation benefit the most from it. Simply put,

before selecting how to best segregate the data using the labels or outputs you've provided, it performs a series of highly complex data transformations.

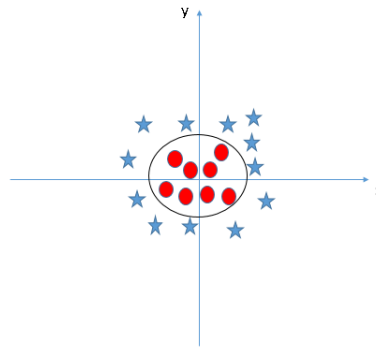


Figure 3. 4: Hyper-plane in original input space looks like a circle

### 3.4.3 Gradient Boosting Decision Tree

The gradient boosting classifier is a boosting ensemble method. Generally speaking, ensemble learning is a model that generates predictions using a variety of models [17]. Because ensemble learning includes several independent models, it is more flexible (less biased) and less data sensitive (less variance). The two most widely used techniques for ensemble learning are bagging and boosting. Boosting is based on the idea of correcting the faults made by the prior learner by instructing the subsequent learner. Gradient Boosting Decision Trees are one example of where boosting is used. Gradient boosting decision trees integrate multiple weak learners into one powerful learner. Individual decision trees in this instance are poor learners. Each tree tries to lessen the error of the one before it as they are connected in a succession. Due to this sequential link, boosting algorithms are often slow to train but incredibly accurate. In statistical learning, slower learning models perform better.

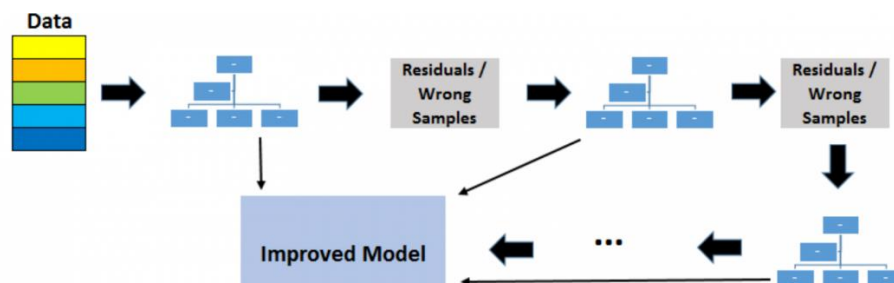


Figure 3. 5: Training of gradient boosting decision tree.

As the figure 4 shown above, the weak learners are fitted in a way that, as the model gets better, each new student fits into the residuals of the stage before them. Each phase's findings are used in the final model to produce a powerful learner. With the use of a loss function, the residuals are found.

### **3.5 Hyperparameter Tuning**

#### **3.5.1 Support Vector Machine (SVM)**

##### **3.5.1.1 Linear Kernel**

When using a linear kernel SVM, the main hyperparameter that needs to be tuned is the regularization parameter  $C$ .

The regularization parameter  $C$  controls the tradeoff between a complex model and a simpler model that may generalize better to new data [18]. A smaller value of  $C$  will result in a wider margin and a simpler model, while a larger value of  $C$  will result in a narrower margin and a more complex model. The best value of  $C$  depends on the particular dataset and problem at hand. Grid search is a popular method for fine-tuning this hyperparameter. Grid search entails creating a grid of  $C$  values and then meticulously assessing the performance of the model for each one. The performance of the model is typically estimated using cross-validation for each value of  $C$ . The final model is then chosen based on the value of  $C$  that provides the highest performance in terms of some evaluation metrics.

For the code, first define a parameter grid that specifies the values of  $C$  to be tried in the grid search. Then create an SVM model with a linear kernel and use the `GridSearchCV` function to perform grid search with cross-validation (for my case, 5-fold cross-validation). The `GridSearchCV` function fits the model for each value of  $C$  and evaluates the model using cross-validation. Finally, print the best hyperparameter and the corresponding score, and use the best hyperparameter to fit the SVM model on the training set.

##### **3.5.1.2 RBF (Radial Basis Function) Kernel**

When using an RBF kernel SVM, there are two main hyperparameters that need to be tuned which are the regularization parameter  $C$  and the kernel coefficient gamma.

RBF kernel SVM is kind of same as Linear kernel SVM, but RBF kernel has an additional hyperparameter which is the kernel coefficient gamma.

The kernel coefficient gamma determines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’ [19]. Intuitively, a small gamma value will consider only close points, while a large gamma value will consider points that are farther apart. The best value of gamma also depends on the particular dataset and problem at hand. For RBF kernel, the method for fine-tuning this hyperparameter is same as Linear kernel which is by using Grid search which involves defining a grid of values for C and gamma.

For the code, first define a parameter grid that specifies the values of C and gamma to be tried in the grid search. Then create an SVM model with an RBF kernel and use the GridSearchCV function to perform grid search with cross-validation (for my case, 5-fold cross-validation). The GridSearchCV function fits the model for each combination of hyperparameters and evaluates the model using cross-validation. Finally, print the best hyperparameters and the corresponding score, and use the best hyperparameters to fit the SVM model on the training set.

### **3.5.2 Gradient Boosting Decision Tree (GDBT)**

There are several hyperparameters in Gradient Boosting Decision Tree that can be tuned to improve the model's performance which are number of trees (n\_estimators), learning rate (learning\_rate) and maximum depth (max\_depth). Number of trees determines the number of decision trees that are built in the model [20]. Increasing the number of trees can improve the model's accuracy, but it also increases the computation time. Learning rate determines the contribution of each tree in the model. A lower learning rate means each tree contributes less to the model, which can improve the model's accuracy, but it also increases the computation time. Maximum depth determines the maximum depth of each decision tree. Increasing the maximum depth can improve the model's accuracy, but it also increases the risk of overfitting. Grid Search Cross Validation is used to hyperparameter tune. In this approach, a grid of hyperparameter values is defined, and the model is trained and evaluated for each combination of hyperparameters using cross-validation. The combination of



hyperparameters that results in the best performance is chosen as the optimal set of hyperparameters.

For the code, initialize the GBDT regressor and define the hyperparameters to tune. Then create an GDBT model and use the GridSearchCV function to perform grid search with cross-validation (for my case, 5-fold cross-validation). Then find the best hyperparameters for the model. We fit the grid search on the training dataset and get the best hyperparameters. Finally, we initialize the GBDT regressor with the best hyperparameters and fit it on the training and testing dataset.

### 3.6 Evaluation

At the end of the modelling, this model is evaluated using several performance metrics such as confusion matrix, F1 score, accuracy and AUC curve. One of the most natural and straightforward methods for figuring out how accurate and precise a model is the confusion matrix [21]. It is employed to address issues of classification if the output can be split into two or more classes. A classification problem prediction outcome summary is known as a confusion matrix [22]. Utilizing count values, the total number of successful and failed predictions is calculated and divided by class. In actuality, the confusion matrix is not a performance statistic in and of itself. However, the majority of performance indicators, including recall, accuracy, and precision, are dependent on the confusion matrix. AUC-ROC curve is the indicator of performance for classification issues at different threshold levels. AUC stands for the level or measurement of separability, and ROC is a probability curve. It reveals how well the model can differentiate across classes. The model is more accurate at classifying 0 classes as 0, and classifying 1 classes as 1, the higher the AUC. The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis. In addition to these performance indicators, a classification report is displayed. It displays accuracy, recall, the F1 score, and model support. It gives us a clearer picture of the overall effectiveness of our trained model.

Formulas:

$$1. \text{ accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$2. \text{ precision} = \frac{TP}{TP+FP}$$

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

$$3. \text{ F1 score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \text{ where recall} = \frac{TP}{TP + FN}$$

where,

$TP = \text{True Positive}, TN = \text{True Negative}, FP = \text{False Positive},$

$FN = \text{False Negative}$

### 3.7 Develop simple Graphic User Interface (GUI)

At the end of the project, a simple GUI for this credit risk prediction is developed. This GUI is developed on the same platform as where the models are trained, which is the anaconda jupyter notebook. Tkinter is the standard GUI library for Python [8]. The default GUI library for Python is called Tkinter. The combination of Python and Tkinter makes it quick and simple to develop GUI apps. An effective object-oriented interface for the Tk GUI toolkit is provided by Tkinter.

Basic steps to develop a simple GUI:

- a) Import the Tkinter module.
- b) Create the GUI application main window.
- c) Add one or more of the above-mentioned widgets to the GUI application.
- d) Enter the main event loop to take action against each event triggered by the user.

## Chapter 4: Experiment/Simulation

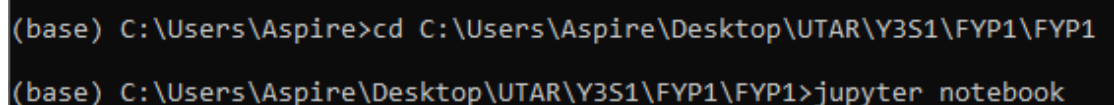
### 4.1 Software Setup

#### 4.1.1 Anaconda and Jupyter Notebook Installation

As stated in Chapter 2, Anaconda and Jupyter Notebook will be used to develop the models and the GUI for credit risk prediction. Anaconda is a free and open-source distribution of Python and R programming languages for scientific computing and data analysis [4]. Jupyter Notebook is a web-based interactive computing environment for creating and sharing documents that contain live code, equations, visualizations, and narrative text. Anaconda and Jupyter Notebook provide a comprehensive and user-friendly environment for data science and scientific computing, making it easy to install, manage, and use the most popular Python packages and tools.

Below are the steps to setup Anaconda and Jupyter Notebook.

- I. Download Anaconda from the official website based on your operating system: <https://www.anaconda.com/products/distribution>
- II. Install Anaconda by following the instructions on the installation wizard.
- III. Once the installation is complete, open the Anaconda Prompt on your device.
- IV. Type "jupyter notebook" and press enter. If you want to open a specific file, open the file by typing `cd "file path"` and press enter, then type "jupyter notebook" and press enter.



```
(base) C:\Users\Aspire>cd C:\Users\Aspire\Desktop\UTAR\Y3S1\FYP1\FYP1
(base) C:\Users\Aspire\Desktop\UTAR\Y3S1\FYP1\FYP1>jupyter notebook
```

Figure 4. 1: Code to open jupyter notebook

- V. This will open Jupyter Notebook in your default web browser. You can now create a new notebook or open an existing one.
- VI. To create a new notebook, click on the "New" button on the right side of the screen and select "Python 3" (or any other kernel that you want to use).

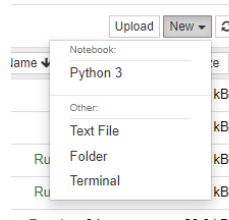


Figure 4. 2: Choose Python 3

- VII. This will open a new notebook where you can write and run Python code.
- VIII. To save your notebook, click on the "File" menu and select "Save and Checkpoint" or use the shortcut "Ctrl + S" (Windows) or "Cmd + S" (Mac).
- IX. To exit Jupyter Notebook, simply close the browser window.

## 4.2 System Implementation

### 4.2.1 Data Exploration

The data is explored. The table below shows the name, type and description of the attributes and whether there is missing value or not.

Attribute Name	Attribute Type	Attribute Description	Missing Value
"LIMIT_BAL"	Numerical	Credit amount (NT dollars)	No
"SEX"	Categorical	Gender, in which 1 indicates "male", while 2 indicates "female".	No
"EDUCATION"	Categorical	Education level, in which 1 indicates "graduate school", 2 indicates "university", 3 indicates "high school", while 4, 5 and 6 indicate "others".	No
"MARRIAGE"	Categorical	Marital status in which 1 indicates "married", 2 indicates "single" and 3 denotes "others".	No
"AGE"	Numerical	Age (years)	No
"PAY_0"	Categorical	"Repayment status" in September of year 2005 in which -1 denotes "pay duly", 1 denotes "payment delay"	No

Chapter 4 Experiment/Simulation

		for one month”, 2 denotes “payment delay for two months”, ... 8 denotes “payment delay for eight months”, 9 denotes “payment delay for nine months and above”.	
“PAY_2”	Categorical	“Repayment status” in August of year 2005. The scales are same as above.	No
“PAY_3”	Categorical	“Repayment status” in July of year 2005. The scales are same as above.	No
“PAY_4”	Categorical	“Repayment status” in June of year 2005. The scales are same as above.	No
“PAY_5”	Categorical	“Repayment status” in May of year 2005. The scales are same as above.	No
“PAY_6”	Categorical	“Repayment status” in April of year 2005. The scales are same as above.	No
“BILL_AMT1”	Numerical	“Bill statement amount” in September of 2005 in “NT dollar”.	No
“BILL_AMT2”	Numerical	“Bill statement amount” in August of 2005 in “NT dollar”.	No
“BILL_AMT3”	Numerical	“Bill statement amount” in July of 2005 in “NT dollar”.	No
“BILL_AMT4”	Numerical	“Bill statement amount” in June of 2005 in “NT dollar”.	No
“BILL_AMT5”	Numerical	“Bill statement amount” in May of 2005 in “NT dollar”.	No
“BILL_AMT6”	Numerical	“Bill statement amount” in April of 2005 in “NT dollar”.	No
“PAY_AMT1”	Numerical	“Previous payment amount” in September of 2005 in “NT	No

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

		dollar”.	
“PAY_AMT2”	Numerical	“Previous payment amount” in August of 2005 in “NT dollar”.	No
“PAY_AMT3”	Numerical	“Previous payment amount” in July of 2005 in “NT dollar”.	No
“PAY_AMT4”	Numerical	“Previous payment amount” in June of 2005 in “NT dollar”.	No
“PAY_AMT5”	Numerical	“Previous payment amount” in May of 2005 in “NT dollar”.	No
“PAY_AMT6”	Numerical	“Previous payment amount” in April of 2005 in “NT dollar”.	No
“default.payment.next.month”	Categorical	“Default payment” in which 1 denotes “yes”, while 0 denotes “no”.	No

Table 4. 1: name, type and description of the attributes and whether there is missing value or not.

As the table above shown, I found that there are missing values in dataset. Hence, data cleaning is not required. The “default.payment.next.month” is changed to default for ease of understanding. All the data are stored in numeric format, but some of the attributes are categorical features. These values of categorical features are converted from numerical into categorical using the `replace()` function, except for the “default” because it will be used as target output variable.

```
credit_card.head(10)
```

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	I
0	1	20000	Female	university	married	24	two	two	duly	duly	...	
1	2	120000	Female	university	single	26	duly	two	zero	zero	...	
2	3	90000	Female	university	single	34	zero	zero	zero	zero	...	
3	4	50000	Female	university	married	37	zero	zero	zero	zero	...	
4	5	50000	Male	university	married	57	duly	zero	duly	zero	...	
5	6	50000	Male	graduate school	single	37	zero	zero	zero	zero	...	

Figure 4. 3: Result of converting numeric value into categoric.

### 4.2.2 Data Visualisation

The data is splitted into input matrix,  $x$  and output vector,  $y$ . The “default” is the output vector and remaining attributes are input matrix. The data is then splitted into training set and test set. Then, the input matrix of training set is splitted into categorical and continuous (numerical) attributes for visualisation. Continuous features are visualized using boxplots and correlation matrix.

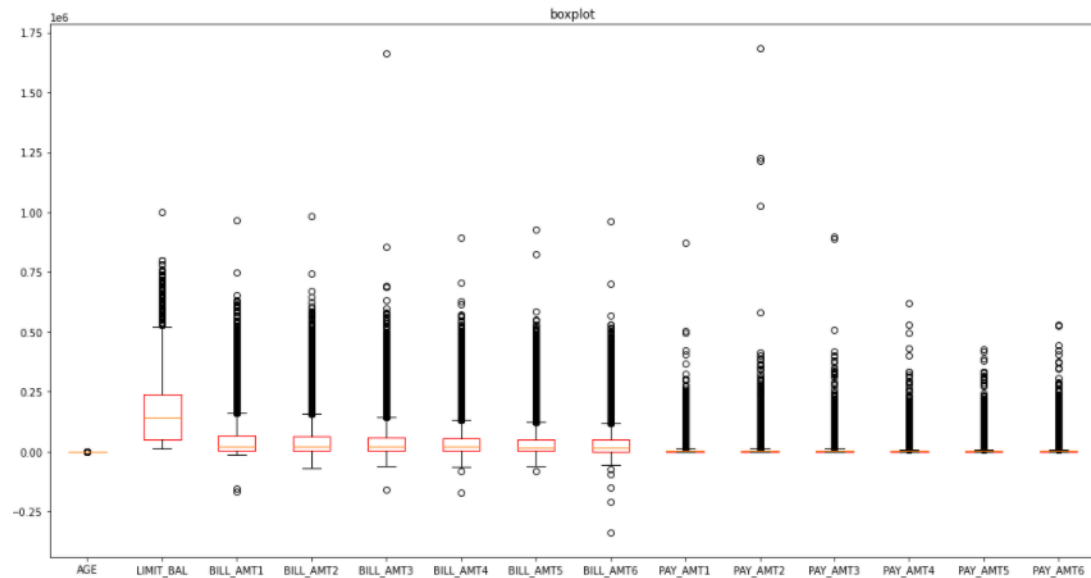


Figure 4. 4: Result of boxplots

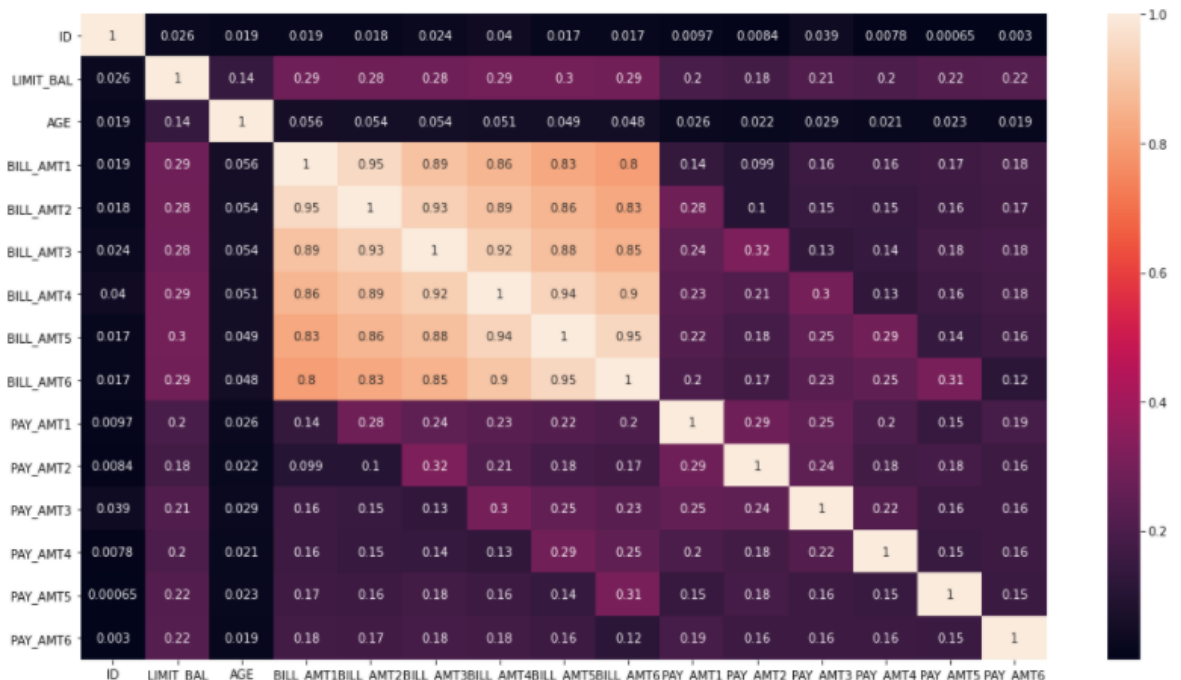


Figure 4. 5: Result of Correlation Matrix

## Chapter 4 Experiment/Simulation

From the correlation matrix, we can see that amount of bill statement from April to September in the year 2005 are highly correlated. Highly correlated features in credit risk assessment might be problematic for a number of reasons. First, they can introduce multicollinearity, which is a condition when two or more features are strongly linked, leading to unstable or challenging-to-interpret coefficients for the features. This can make figuring out which characteristics are actually crucial for predicting credit risk difficult. Besides that, strongly correlated characteristics have the potential to introduce redundancy, which means that some features may not add much to the information already being recorded by other features. This may diminish the model's interpretability and increase the computing cost of the research.

The categorical features are visualized using contingency table.

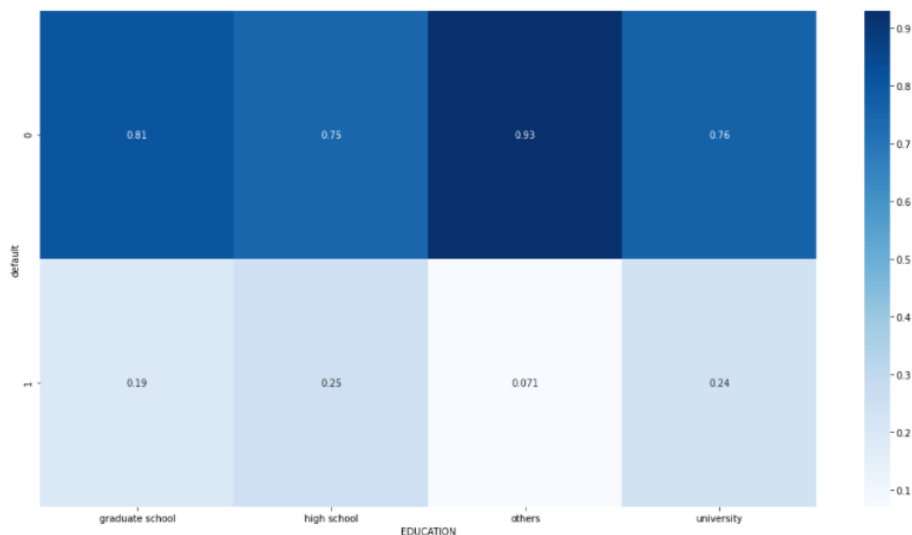


Figure 4. 6: Result of contingency table

From the contingency table, we can deduce that the class distribution is imbalanced, with higher proportion of non-default clients. Thus, accuracy might not be appropriate when it comes to model's performance evaluation.

You can visualize the relationship between other categorical features and target by changing feature name. As you can see from Figure 4.5 the value default data which are "0" and "1" has a large difference in proportions. This means that the data is extremely imbalanced which may affect the accuracy of the prediction.



### 4.2.3 Feature Engineering

During the data exploration, I found that the data in some attributes has large range such as “LIMIT\_BAL” and “AGE”. Categorical feature encoding is done using the using the `get_dummies()` function from Pandas library.

```
credit_card1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 89 columns):
```

Figure 4. 7: Result of encoding

As you can see in Figure 4.6, the number of columns increased to 89 after the encoding.

Normalisation of data is done using min-max normalisation to convert the values to a common scale.

```
from sklearn import preprocessing as prep
minmax_scale = prep.MinMaxScaler().fit(credit_card1)
credit_minmax = minmax_scale.transform(credit_card1)
credit_minmax = pd.DataFrame(credit_minmax, columns = list(credit_card1))
credit_minmax
```

	ID	LIMIT_BAL	AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	...	PAY_6_duly	PAY_6_eight	PAY_6_five	PAY_6_four
0	0.000000	0.010101	0.051724	0.149982	0.069164	0.086723	0.160138	0.080648	0.260979	0.000000	...	0.0	0.0	0.0	0.0
1	0.000033	0.111111	0.086207	0.148892	0.067858	0.087817	0.163220	0.084074	0.263485	0.000000	...	0.0	0.0	0.0	0.0
2	0.000067	0.080808	0.224138	0.172392	0.079532	0.093789	0.173637	0.095470	0.272928	0.001738	...	0.0	0.0	0.0	0.0
3	0.000100	0.040404	0.275862	0.188100	0.111995	0.113407	0.188809	0.109363	0.283685	0.002290	...	0.0	0.0	0.0	0.0
4	0.000133	0.040404	0.620690	0.154144	0.071601	0.106020	0.179863	0.099633	0.275681	0.002290	...	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29995	0.999867	0.212121	0.310345	0.313716	0.249208	0.200746	0.243036	0.111622	0.273259	0.009730	...	0.0	0.0	0.0	0.0
29996	0.999900	0.141414	0.379310	0.148008	0.067955	0.088267	0.168596	0.085794	0.260979	0.002103	...	0.0	0.0	0.0	0.0
29997	0.999933	0.020202	0.275862	0.149674	0.069405	0.087859	0.179805	0.101057	0.275854	0.000000	...	0.0	0.0	0.0	0.0
29998	0.999967	0.070707	0.344828	0.145064	0.140604	0.128239	0.209850	0.092403	0.298591	0.098334	...	1.0	0.0	0.0	0.0
29999	1.000000	0.040404	0.431034	0.188931	0.112633	0.113667	0.194553	0.112803	0.272746	0.002379	...	0.0	0.0	0.0	0.0

Figure 4. 8: Result of normalization

The default data of this dataset is found to be imbalanced, and this is proven from the Figure 4.5. Hence, Synthetic Minority Oversampling Technique (SMOTE) will be implemented to overcome this problem. Synthetic Minority Oversampling Technique (SMOTE) approach. SMOTE is an algorithm that adds artificial data points to the actual data points to accomplish data augmentation. SMOTE can be viewed as an improved form of oversampling or as a particular data augmentation procedure. With SMOTE, you avoid producing duplicate data points and instead produce synthetic data points that are marginally different from the original data points. This SMOTE approach will increase the sample of default class so that the data will be more

balanced. In other words, the SMOTE approach will help to boost the accuracy of the model.

```
: X1 = credit_minmax.drop(["default"],axis=1)
  y1 = credit_minmax["default"]

: # Instantiate a SMOTE object
  sm = SMOTE(random_state=42)

: # Use SMOTE to resample the data
  X_res, y_res = sm.fit_resample(X1, y1)

: # Print the number of samples before and after SMOTE
  print("Number of samples before SMOTE:", len(X))
  print("Number of samples after SMOTE:", len(X_res))

Number of samples before SMOTE: 30000
Number of samples after SMOTE: 46728
```

Figure 4. 9: Result of sampling data with SMOTE approach

## 4.2.4 Model Training

After the data preparation and feature engineering is completed, the data is modelled using the Logistic Regression, Support Vector Machine and Gradient Boosting Decision Tree algorithm.

### 4.2.4.1 Logistic Regression

```
X1_train, X1_test, y1_train, y1_test = train_test_split(X_res, y_res, test_size=0.2, random_state=30)

print("X1_train: ", X1_train.shape)
print("y1_train: ", y1_train.shape)
print("X1_test: ", X1_test.shape)
print("y1_test: ", y1_test.shape)

X1_train: (37382, 87)
y1_train: (37382,)
X1_test: (9346, 87)
y1_test: (9346,)
```

Figure 4. 10: Split the dataset into training and testing data

Split the dataset into training and testing data using the “train\_test\_split” function from scikit-learn. Here, we are setting the “test\_size” parameter to 0.2, which means that 20% of the data will be used for testing and the rest 80% will be used for training. The random\_state parameter is set to 30 to ensure reproducibility of the results.

We also split the features, X and the target variable y into their respective training and testing sets.

```
from sklearn.linear_model import LogisticRegression
accuracy={}
model = LogisticRegression()
model.fit(X1_train, y1_train)

test_pred = model.predict(X1_test)
```

Figure 4. 11: Train the data with Logistic Regression algorithm

Next, create a logistic regression model using the `LogisticRegression` class from `scikit-learn`. By default, this creates a logistic regression model with L2 regularization. Finally, train the logistic regression model on the training data using the `fit` method. This updates the coefficients of the logistic regression model based on the training data.

After training the model, use it to make predictions on new data using the `predict` method.

#### 4.2.4.2 Support Vector Machine (SVM)

```
X1_train, X1_test, y1_train, y1_test = train_test_split(X_res, y_res, test_size=0.2, random_state=30)

print("X1_train: ", X1_train.shape)
print("y1_train: ", y1_train.shape)
print("X1_test: ", X1_test.shape)
print("y1_test: ", y1_test.shape)

X1_train:  (37382, 87)
y1_train:  (37382,)
X1_test:   (9346, 87)
y1_test:   (9346,)
```

Figure 4. 12: Split the dataset into training and testing data

Split the dataset into training and testing data using the “`train_test_split`” function from `scikit-learn`. Here, we are setting the “`test_size`” parameter to 0.2, which means that 20% of the data will be used for testing and the rest 80% will be used for training. The `random_state` parameter is set to 30 to ensure reproducibility of the results.

We also split the features, X and the target variable y into their respective training and testing sets.

For SVM model, Linear Kernel and Radial Basis Function (RBF) Kernel will be used to train the model.

##### 4.2.4.2.1 Linear Kernel SVM

```
: from sklearn.svm import SVC
  ##Linear Kernel SVM
  # Train the SVM model
  svm = SVC(kernel='linear', random_state=42)
  svm.fit(X1_train, y1_train)

  # Make predictions on the testing set
  y_pred = svm.predict(X1_test)
```

Figure 4. 13: Train the data with Linear Kernel SVM algorithm

Next, create a SVM model with a linear kernel and random state of 42 using the SVC class from scikit-learn. Finally, train the Linear Kernel SVM model on the training data using the fit method.

After training the model, use it to make predictions on new data using the predict method.

#### 4.2.4.2.1 Radial Basis Function (RBF) Kernel SVM

```
X1_train, X1_test, y1_train, y1_test = train_test_split(X_res, y_res, test_size=0.2, random_state=30)

print("X1_train: ", X1_train.shape)
print("y1_train: ", y1_train.shape)
print("X1_test: ", X1_test.shape)
print("y1_test: ", y1_test.shape)

X1_train: (37382, 87)
y1_train: (37382,)
X1_test: (9346, 87)
y1_test: (9346,)
```

Figure 4. 14: Split the dataset into training and testing data

Split the dataset into training and testing data using the “train\_test\_split” function from scikit-learn. Here, we are setting the “test\_size” parameter to 0.2, which means that 20% of the data will be used for testing and the rest 80% will be used for training. The random\_state parameter is set to 30 to ensure reproducibility of the results.

We also split the features, X and the target variable y into their respective training and testing sets.

```
##RBF Kernel SVM
# Train the SVM with RBF kernel
svm = SVC(kernel='rbf', random_state=42)
svm.fit(X1_train, y1_train)

# Make predictions on the test set
y1_pred = svm.predict(X1_test)
```

Figure 4. 15: Train the data with RBF Kernel SVM algorithm

Next, create a SVM model with a rbf kernel and random state of 42 using the SVC class from scikit-learn. Finally, train the RBF Kernel SVM model on the training data using the fit method.

After training the model, use it to make predictions on new data using the predict method.

### 4.2.4.3 Gradient Boosting Decision Tree (GBDT)

```
X1_train, X1_test, y1_train, y1_test = train_test_split(X_res, y_res, test_size=0.2, random_state=30)

print("X1_train: ", X1_train.shape)
print("y1_train: ", y1_train.shape)
print("X1_test: ", X1_test.shape)
print("y1_test: ", y1_test.shape)

X1_train: (37382, 87)
y1_train: (37382,)
X1_test: (9346, 87)
y1_test: (9346,)
```

Figure 4. 16: Split the dataset into training and testing data

Split the dataset into training and testing data using the “train\_test\_split” function from scikit-learn. Here, we are setting the “test\_size” parameter to 0.2, which means that 20% of the data will be used for testing and the rest 80% will be used for training. The random\_state parameter is set to 30 to ensure reproducibility of the results.

We also split the features, X and the target variable y into their respective training and testing sets.

```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
# Instantiate a GradientBoostingClassifier
gb_clf = GradientBoostingClassifier(n_estimators=300, learning_rate=1, max_depth=5, random_state=42)

# Fit the classifier to the training data
gb_clf.fit(X1_train, y1_train)

# Make predictions on the testing data
y_pred = gb_clf.predict(X1_test)
```

Figure 4. 17: Train the data with GBDT algorithm

Next, we create a Gradient Boosting Decision Tree model using the Gradient Boosting classifier from scikit-learn with some hyperparameters. Set the random state to 42, the number of estimators or the number of decision trees to 300, the maximum depth of each tree to 5, and the learning rate to 1.

We then fit the classifier on the training dataset. This trains the Gradient Boosting Decision Tree model by sequentially adding decision trees to minimize the loss function. Loss function is the objective function that measures the difference between the predicted and actual values.

After training the model, use it to make predictions on new data using the predict method.

**4.2.5 Hyperparameter Tuning****4.2.5.1 Support Vector Machine (SVM)****4.2.5.1.1 Linear Kernel**

```

: from sklearn.model_selection import GridSearchCV
  #LINEAR parameter fine-tuning
  # Train the SVM model
  svm = SVC(kernel='linear', random_state=42)

  # Define a range of hyperparameters to test
  param_grid = {
      'C': [0.1, 1, 10, 100],
      'class_weight': [None, 'balanced']
  }

  # Perform grid search to find the best parameters
  grid_search = GridSearchCV(svm, param_grid, scoring='f1', cv=5)
  grid_search.fit(Xs_train, ys_train)

  # Print the best parameters and score
  print('Best Parameters:', grid_search.best_params_)
  print('Best Score:', grid_search.best_score_)

Best Parameters: {'C': 1, 'class_weight': 'balanced'}
Best Score: 0.5092454649369543

```

```

# Make predictions on the test set using the best model
best_svm = grid_search.best_estimator_
ys_pred = best_svm.predict(Xs_test)

```

Figure 4. 18: Hyperparameter tuning the Linear Kernel SVM

Create an SVM model with a linear kernel using the SVC class from scikit-learn's svm module. Then define a hyperparameter grid to search over, consisting of different values of “C” (0.1, 1, 10 and 100) and “class\_weight”, where 'None' means no class weighting and 'balanced' means using a balanced class weighting strategy.

We perform a grid search using 5-fold cross-validation to find the best hyperparameters for the SVM model and fit it on the training set using the fit() method of the GridSearchCV object. Finally, we print the best hyperparameters found by the grid search and the corresponding accuracy on the test set.

After that, use it to make predictions on new data using the predict method.

**4.2.5.1.2 Radial Basis Function (RBF) Kernel**

```

#RBF parameter fine-tuning
# Define the SVM model and parameter grid for grid search
svm = SVC(kernel='rbf', random_state=42)
param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma': [0.01, 0.1, 1, 10]
}

# Perform grid search to find the best parameters
grid_search = GridSearchCV(svm, param_grid, scoring='f1', cv=5)
grid_search.fit(Xs_train, ys_train)

# Print the best parameters and score
print('Best Parameters:', grid_search.best_params_)
print('Best Score:', grid_search.best_score_)

# Make predictions on the test set using the best model
best_svm = grid_search.best_estimator_
ys2_pred = best_svm.predict(Xs_test)

```

Figure 4. 19: Hyperparameter tuning the RBF Kernel SVM

Create an SVM model with an RBF kernel using the SVC class from scikit-learn's svm module. We then define a hyperparameter grid to search over, consisting of different values of “C” (0.1, 1, 10 and 100) and “gamma” (0.01, 0.1, 1, 10).

We perform a grid search using 5-fold cross-validation to find the best hyperparameters for the SVM model and fit it on the training set using the fit() method of the GridSearchCV object. Finally, we print the best hyperparameters found by the grid search and the corresponding accuracy on the test set.

After that, use it to make predictions on new data using the predict method.

### 4.2.5.2 Gradient Boosting Decision Tree (GBDT)

```

from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Instantiate a GradientBoostingClassifier
gb_clf = GradientBoostingClassifier(random_state=42)

# Define the hyperparameter grid
param_grid = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.1, 1],
    'max_depth': [3, 4, 5]
}

# Instantiate a GridSearchCV object
grid_search = GridSearchCV(estimator=gb_clf, param_grid=param_grid, cv=5, n_jobs=-1)

# Fit the GridSearchCV object to the training data
grid_search.fit(X1_train, y1_train)

# Print the best hyperparameters
print("Best Parameters: ", grid_search.best_params_)

# Make predictions on the testing data using the best estimator
best_clf = grid_search.best_estimator_
y1_pred = best_clf.predict(X1_test)

```

Figure 4. 20: Hyperparameter tuning the GBDT model

Set up the hyperparameters to tune in a dictionary “param\_grid”. Specify the ranges of values for the hyperparameters “n\_estimators”, “learning\_rate”, “max\_depth”, and fix the value of random\_state to ensure reproducibility.

We then initialize the GBDT classifier with default hyperparameters and set up the GridSearchCV object with 5-fold cross-validation. The GridSearchCV object searches over all possible combinations of hyperparameters in “param\_grid” to find the best hyperparameters.

We fit the GridSearchCV object on the training dataset, which trains and validates the GBDT models with different hyperparameters. After the search, we print the best hyperparameters that maximize the scoring metric.

Next, we use the best model obtained from the hyperparameter search to predict the credit risk in the testing dataset using predict method.

### 4.2.6 Performance Evaluation

After the model training, these models will be evaluated using accuracy, precision, recall, f1 score, AUC-ROC curve, confusion matrix and classification report. The result of the evaluation will be further discussed in Chapter 6.

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR



```
# Evaluate the performance of the classifier using the best hyperparameters
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
print("Accuracy:", accuracy_score(y1_test, y1_pred))
print("Precision:", precision_score(y1_test, y1_pred))
print("Recall:", recall_score(y1_test, y1_pred))
print("F1 Score:", f1_score(y1_test, y1_pred))
```

Figure 4. 21: Coding of Accuracy, Precision, Recall &amp; F1 Score

```
: from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(y1_test, y1_pred))
print(classification_report(y1_test, y1_pred))
skplt.metrics.plot_confusion_matrix(y1_test, y_pred)
plt.show()
```

Figure 4. 22: Coding of Confusion Matrix &amp; Classification Report

```
# Suppose you have actual y_test values and predicted probabilities y_pred_proba
fpr, tpr, thresholds = roc_curve(y1_test, y1_pred)

# Compute the area under the ROC curve
roc_auc = auc(fpr, tpr)

# Plot the ROC curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (AUC = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.show()
```

Figure 4. 23: Coding of AUC-ROC curve

#### 4.2.7 Develop simple Graphic User Interface (GUI)

After the evaluation of the models, we move on to model selection based on the results of the evaluation. Model with highest accuracy and the best precision, recall and f1 score will be chosen. Create a Graphic User Interface (GUI) on **Jupyter Notebook** using **Tkinter** and implement the model into the GUI with the help of **jolib** library to predict whether the borrower has defaulted on a loan or not. Below is the picture of the GUI created. After user input all the details, they can click the predict button, a message will appear at the bottom right showing the result of the prediction. The details of the borrower will be saved in a text file with the borrower's name.

## Chapter 4 Experiment/Simulation

**Credit Risk Prediction**

NAME:  BILL AMOUNT 1:  Notes: For education, input 0 for others, 1 for graduate school, 2 for university, 3 for high school.

LIMIT BALANCE:  BILL AMOUNT 2:  For marriage, input 0 for others, 1 for married, 2 for single.

Sex (1-Male/2-Female):  BILL AMOUNT 3:  For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.

Education:  BILL AMOUNT 4:

Marriage:  BILL AMOUNT 5:

Age:  BILL AMOUNT 6:

Repayment status 1:  PAY AMOUNT 1:

Repayment status 2:  PAY AMOUNT 2:

Repayment status 3:  PAY AMOUNT 3:

Repayment status 4:  PAY AMOUNT 4:

Repayment status 5:  PAY AMOUNT 5:

Repayment status 6:  PAY AMOUNT 6:

Figure 4. 24: GUI for credit risk prediction

**Credit Risk Prediction**

NAME:  BILL AMOUNT 1:  Notes: For education, input 0 for others, 1 for graduate school, 2 for university, 3 for high school.

LIMIT BALANCE:  BILL AMOUNT 2:  For marriage, input 0 for others, 1 for married, 2 for single.

Sex (1-Male/2-Female):  BILL AMOUNT 3:  For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.

Education:  BILL AMOUNT 4:

Marriage:  BILL AMOUNT 5:

Age:  BILL AMOUNT 6:

Repayment status 1:  PAY AMOUNT 1:

Repayment status 2:  PAY AMOUNT 2:

Repayment status 3:  PAY AMOUNT 3:

Repayment status 4:  PAY AMOUNT 4:

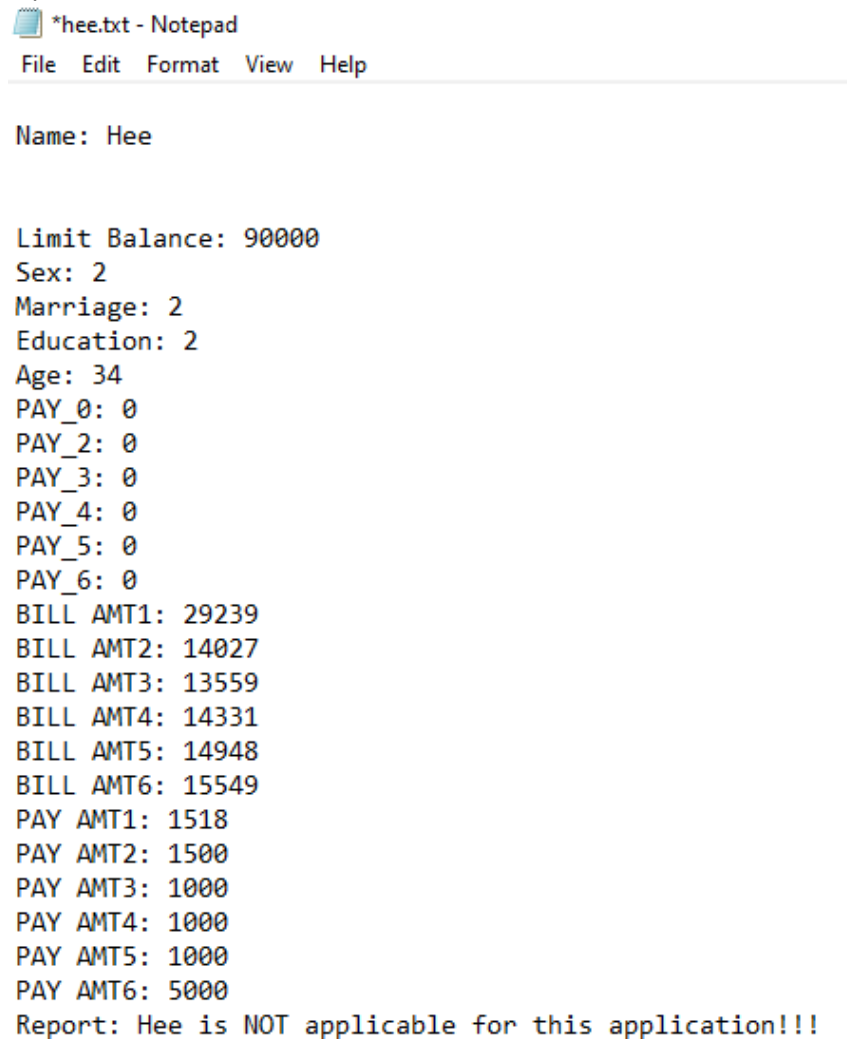
Repayment status 5:  PAY AMOUNT 5:

Repayment status 6:  PAY AMOUNT 6:

**hee not defaulted on loan!!!**

hee's credit risk prediction report is saved in hee.txt

Figure 4. 25: Example result of prediction



```
*hee.txt - Notepad
File Edit Format View Help

Name: Hee

Limit Balance: 90000
Sex: 2
Marriage: 2
Education: 2
Age: 34
PAY_0: 0
PAY_2: 0
PAY_3: 0
PAY_4: 0
PAY_5: 0
PAY_6: 0
BILL AMT1: 29239
BILL AMT2: 14027
BILL AMT3: 13559
BILL AMT4: 14331
BILL AMT5: 14948
BILL AMT6: 15549
PAY AMT1: 1518
PAY AMT2: 1500
PAY AMT3: 1000
PAY AMT4: 1000
PAY AMT5: 1000
PAY AMT6: 5000
Report: Hee is NOT applicable for this application!!!
```

Figure 4. 26: Prediction detail saved in text file

### 4.3 Implementation Issues and Challenges

During the implementation phase, I was faced with a problem when hyperparameter tuning the SVM (Linear kernel & RBF kernel) models. Due to low specification of laptop, the running time of hyperparameter tuning code is longer than expected. Large number of samples (30000) cause it to take a long time to get the results, up to a day. SVM models are known to have a relatively high computational cost compared to other machine learning algorithms, especially when working with large datasets so it will take up more memory and more time for the mathematical calculation hence increasing the training time. To solve this problem, I have to reduce the number of samples by random sampling.

#### **4.4 Concluding Remark**

All the necessary hardware setup and software setup were explained in detail. The system operation was shown and explained to show how to train the model and a brief look at the GUI created.

The GUI has limited use case as it can only be utilized for prediction if we are aware of all the relevant features used to develop the model.

## Chapter 5: System Evaluation and Discussion

### 5.1 System Testing and Performance Evaluation of Models

#### 5.1.1 Accuracy, Precision, Recall & F1 score

```
Accuracy: 0.7037235180826021
Precision: 0.7708215297450425
Recall: 0.5812860499893185
F1 Score: 0.6627694556083302
```

Figure 5. 1: Test Accuracy, Precision, Recall & F1 score of Logistic Regression model

```
Accuracy score: 0.6839289535630216
Precision score: 0.775438596491228
Recall score: 0.5193334757530442
F1 score: 0.6220573183213919
```

Figure 5. 2: Test Accuracy, Precision, Recall & F1 score of Linear kernel SVM model

```
Accuracy: 0.7825
Precision: 0.48161764705882354
Recall: 0.5219123505976095
F1 Score: 0.5009560229445507
```

Figure 5. 3: Test Accuracy, Precision, Recall & F1 score of Linear kernel SVM model after hyperparameter tuning

```
Accuracy: 0.7312219131179114
Precision: 0.811548405630566
Recall: 0.6035035248878444
F1 Score: 0.6922322960058809
```

Figure 5. 4: Test Accuracy, Precision, Recall & F1 score of RBF kernel SVM model

```
Accuracy: 0.8225
Precision: 0.6979166666666666
Recall: 0.26693227091633465
F1 Score: 0.38616714697406335
```

Figure 5. 5: Test Accuracy, Precision, Recall & F1 score of RBF kernel SVM model after hyperparameter tuning

```
Accuracy: 0.841857479135459  
Precision: 0.8594837261503928  
Recall: 0.8179876094851527  
F1 Score: 0.8382224168126094
```

Figure 5. 6: Test Accuracy, Precision, Recall & F1 score of GBDT model

```
Accuracy: 0.86347100363792  
Precision: 0.9117291414752116  
Recall: 0.8053834650715659  
F1 Score: 0.8552631578947368
```

Figure 5. 7: Test Accuracy, Precision, Recall & F1 score of GBDT model after hyperparameter tuning

For the model training I have chosen 3 models which are Logistic Regression, Support Vector Machine and Gradient Boost Decision Tree. Firstly, the Logistic Regression, Support Vector Machine and Gradient Boost Decision Tree models performed decently on the given dataset. The test accuracy for all the 3 models were over 60%. The test accuracy for the 3 models were quite different. The same models also gave a different accuracy score after the hyperparameter tuned them. The accuracy of Linear kernel SVM model increases from 68.39% to 78.25%, accuracy of RBF kernel SVM model increases from 73.12% to 82.25% and accuracy of Gradient Boosting Decision Tree model increases from 84.19% to 86.35%. Obviously, the accuracy of the models is boosted after the hyperparameter tuning. The one that performed best here is the Gradient Boosting Decision Tree model with the highest accuracy of 86.35%.

**5.1.2 Classification report & Confusion matrix**

	precision	recall	f1-score	support
0.0	0.83	0.66	0.74	5816
1.0	0.58	0.77	0.66	3530
accuracy			0.70	9346
macro avg	0.70	0.72	0.70	9346
weighted avg	0.73	0.70	0.71	9346

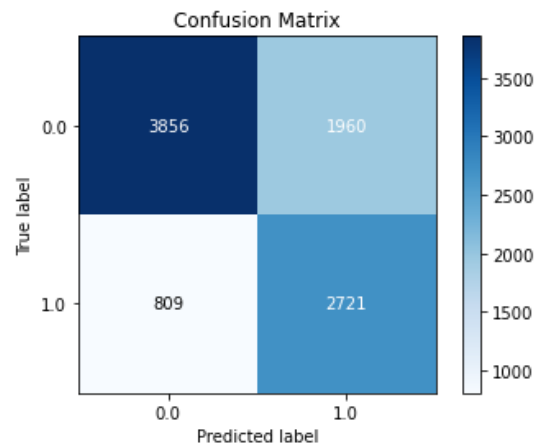


Figure 5. 8: Classification report &amp; Confusion matrix of Logistic Regression model

	precision	recall	f1-score	support
0.0	0.64	0.85	0.73	4665
1.0	0.78	0.52	0.62	4681
accuracy			0.68	9346
macro avg	0.71	0.68	0.68	9346
weighted avg	0.71	0.68	0.68	9346

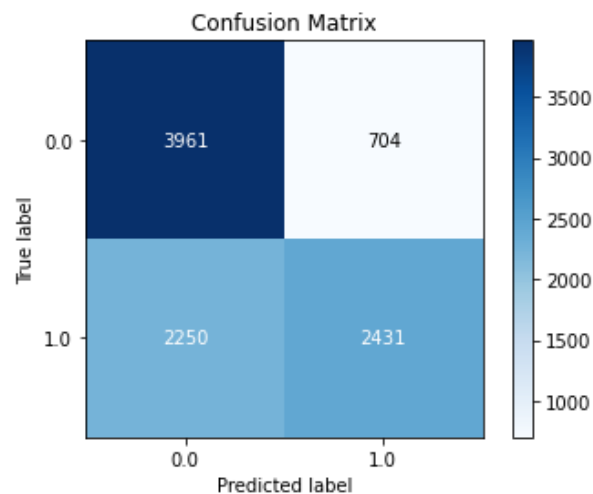


Figure 5. 9: Classification report &amp; Confusion matrix of Linear kernel SVM model

## Chapter 5 System Evaluation and Discussion

	precision	recall	f1-score	support
0.0	0.87	0.85	0.86	949
1.0	0.48	0.52	0.50	251
accuracy			0.78	1200
macro avg	0.68	0.69	0.68	1200
weighted avg	0.79	0.78	0.79	1200

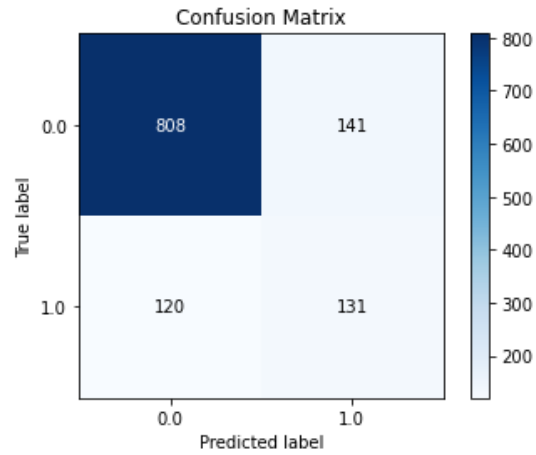


Figure 5. 10: Classification report & Confusion matrix of Linear kernel SVM model after hyperparameter tuning

	precision	recall	f1-score	support
0.0	0.68	0.86	0.76	4665
1.0	0.81	0.60	0.69	4681
accuracy			0.73	9346
macro avg	0.75	0.73	0.73	9346
weighted avg	0.75	0.73	0.73	9346

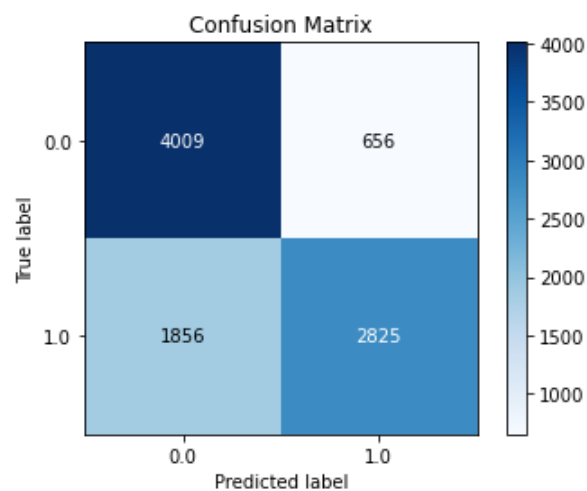


Figure 5. 11: Classification report & Confusion matrix of RBF kernel SVM model



	precision	recall	f1-score	support
0.0	0.83	0.97	0.90	949
1.0	0.70	0.27	0.39	251
accuracy			0.82	1200
macro avg	0.77	0.62	0.64	1200
weighted avg	0.81	0.82	0.79	1200

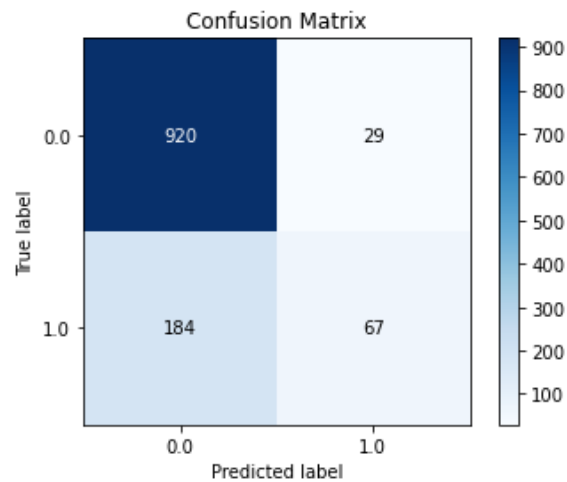


Figure 5. 12: Classification report & Confusion matrix of RBF kernel SVM model after hyperparameter tuning

	precision	recall	f1-score	support
0.0	0.83	0.87	0.85	4665
1.0	0.86	0.82	0.84	4681
accuracy			0.84	9346
macro avg	0.84	0.84	0.84	9346
weighted avg	0.84	0.84	0.84	9346

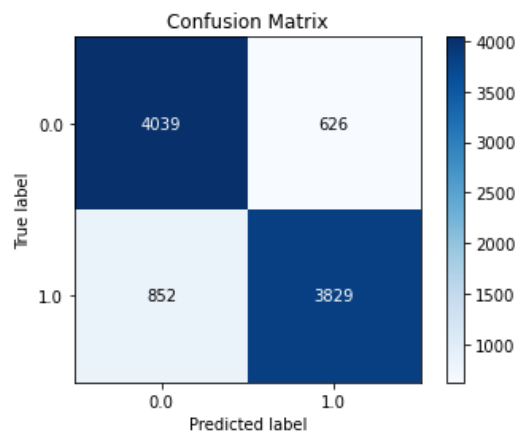


Figure 5. 13: Classification report & Confusion matrix of Gradient Boosting Decision Tree model

	precision	recall	f1-score	support
0.0	0.83	0.92	0.87	4665
1.0	0.91	0.81	0.86	4681
accuracy			0.86	9346
macro avg	0.87	0.86	0.86	9346
weighted avg	0.87	0.86	0.86	9346

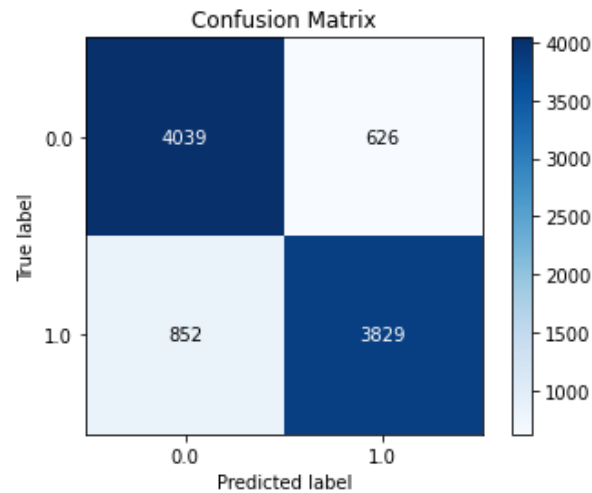


Figure 5. 14: Classification report & Confusion matrix of Gradient Boosting Decision Tree model after hyperparameter tuning

The overall precision, recall and f1 score of the three models are decent with over 60 percents. However, there are still some models having low precision and recall on finding the borrower who has defaulted on loan. For example, Logistic Regression model has low precision and Liner kernel SVM model has low recall on finding the borrower who has defaulted on loan. Low precision indicates that the model is not very reliable in predicting positive instances. A large proportion of instances predicted as positive are actually negative. In other words, the classifier is making many false positive predictions. Low recall means the model is not very effective in identifying positive instances. A large proportion of actual positive instances are predicted as negative. In other words, the classifier is missing many positive instances. Among all these three models, Gradient Boosting Decision Tree has the highest precision, recall and f1 score. In fact, the precision and f1 score are boosted even more after hyperparameter tuning the Gradient Boosting Decision Tree model. This indicates that GBDT model has the best performance compared to the other two models.

### 5.1.1 AUC-ROC curve

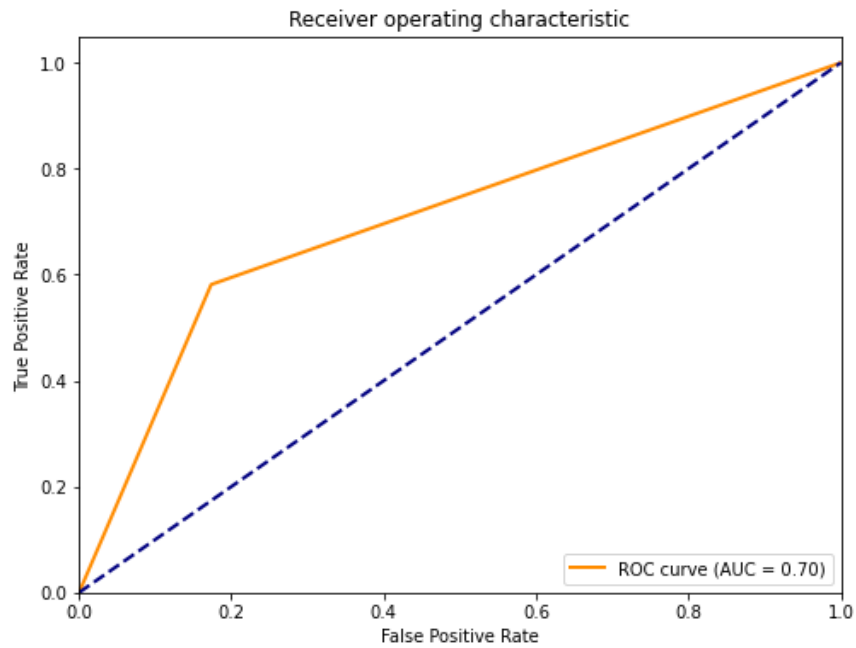


Figure 5. 15: AUC-ROC curve of Logistic Regression model

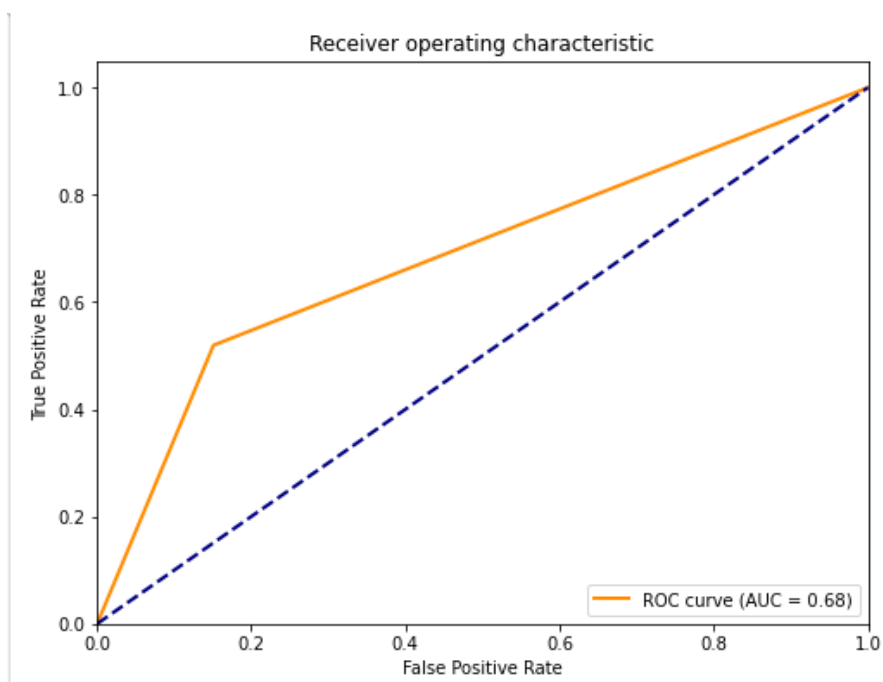


Figure 5. 16: AUC-ROC curve of Linear kernel SVM model

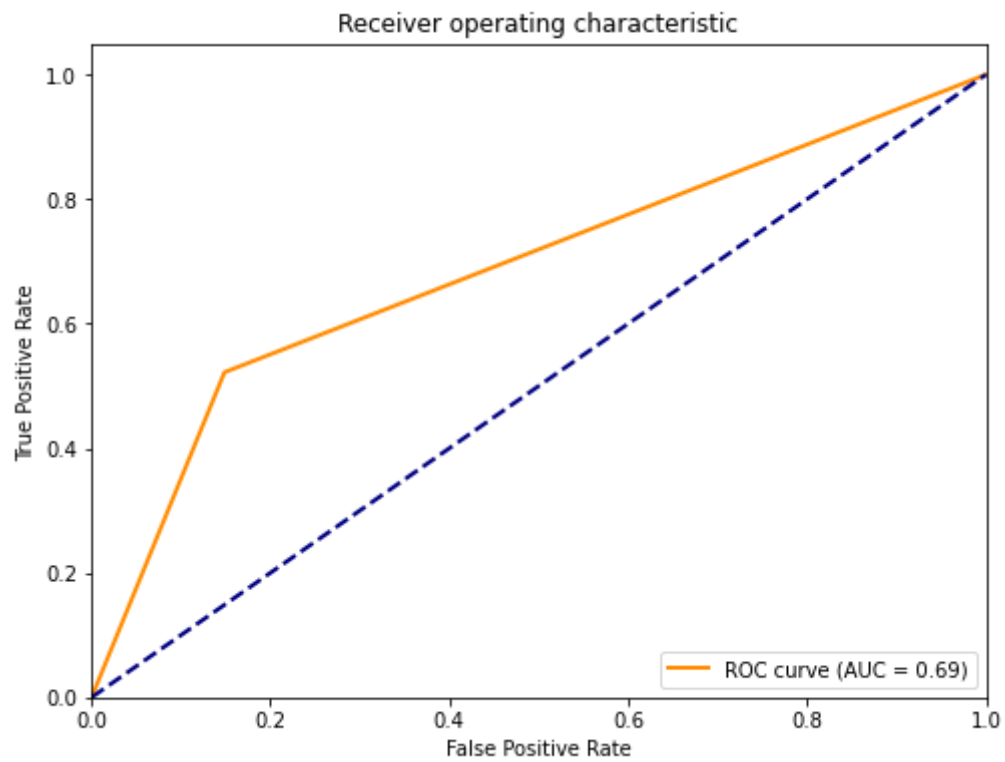


Figure 5. 17: AUC-ROC curve of Linear kernel SVM model after hyperparameter tuning

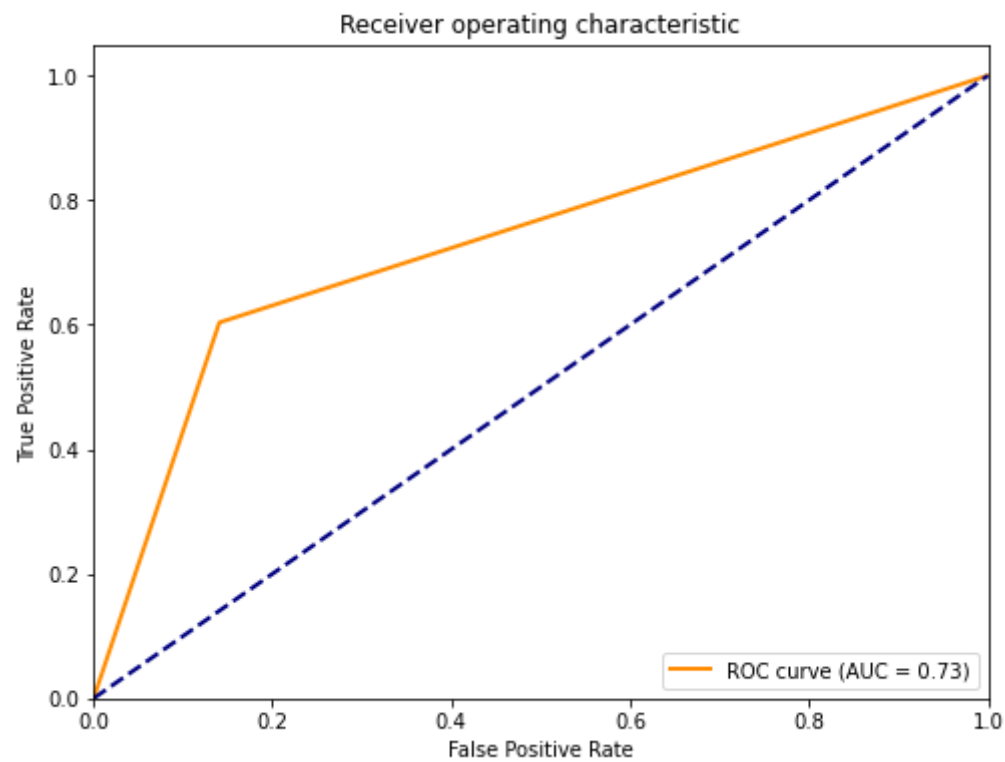


Figure 5. 18: AUC-ROC curve of RBF kernel SVM model

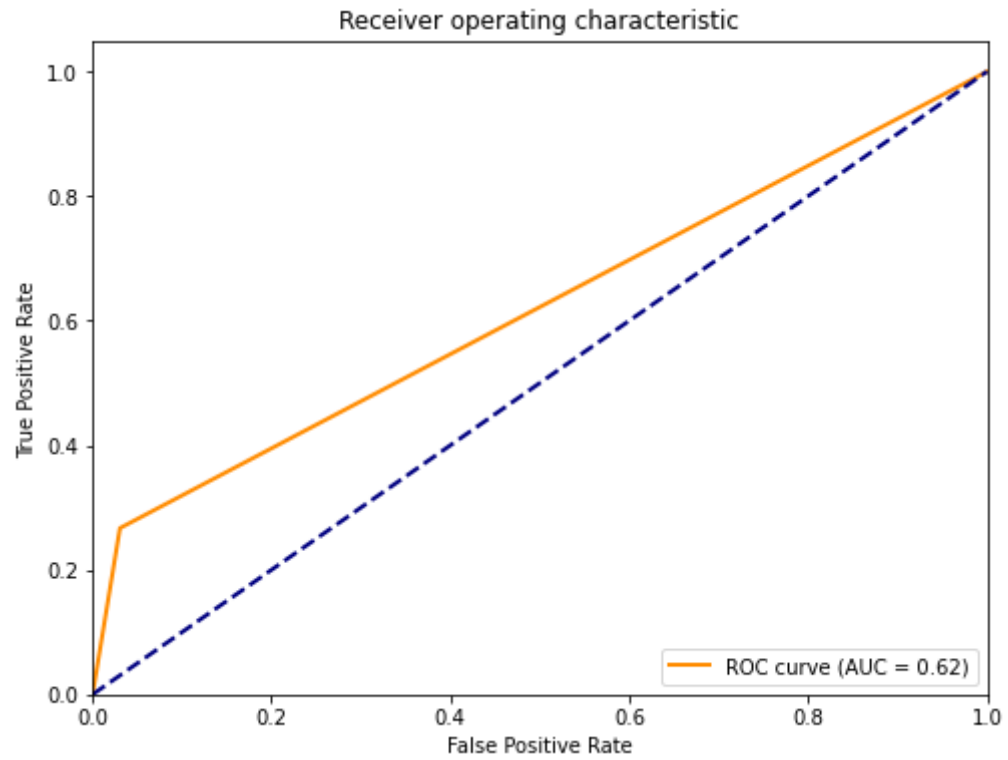


Figure 5. 19: AUC-ROC curve of RBF kernel SVM model after hyperparameter tuning

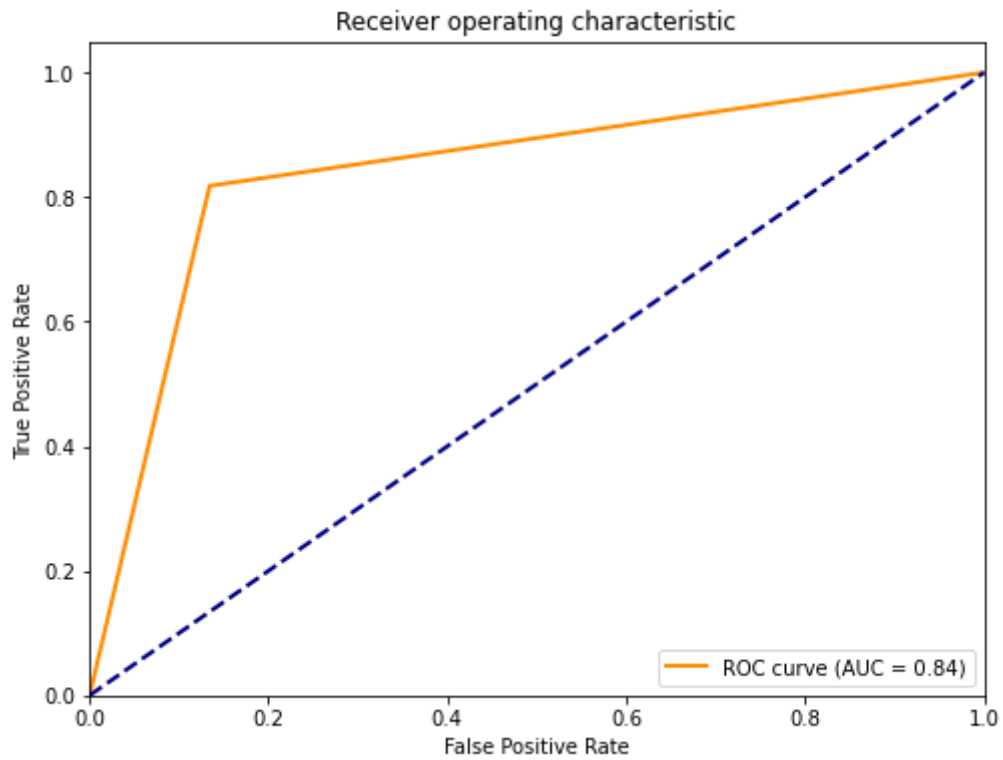


Figure 5. 20: AUC-ROC curve of Gradient Boosting Decision Tree model

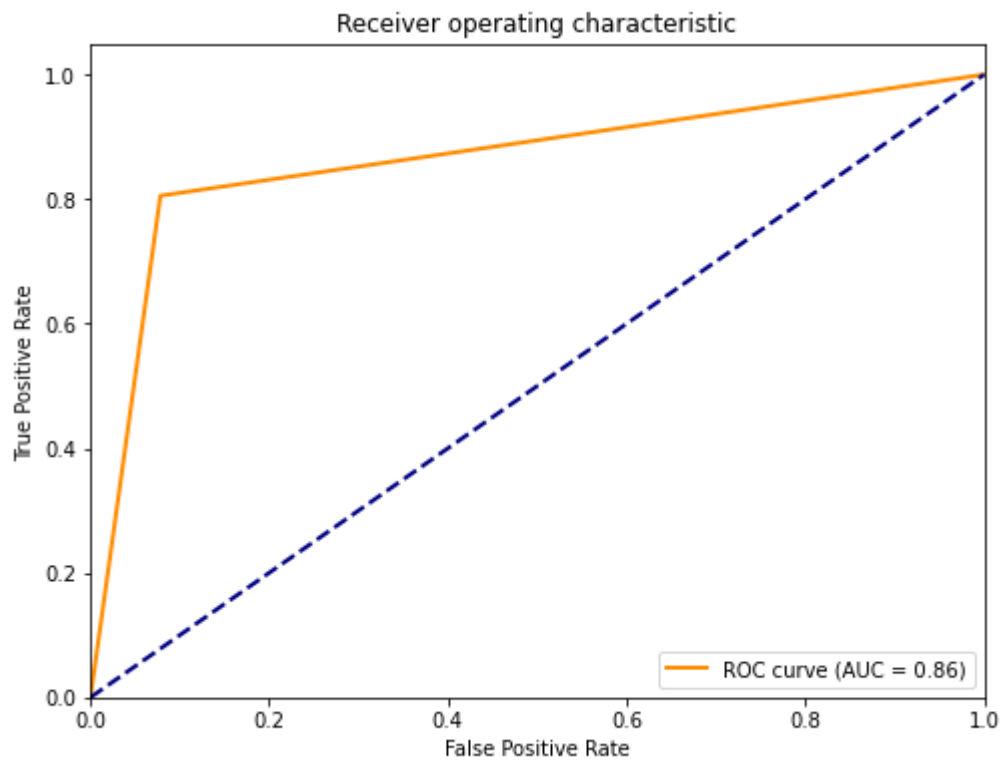


Figure 5. 21: AUC-ROC curve of Gradient Boosting Decision Tree model after hyperparameter tuning

The AUC (Area Under the Curve) of the ROC curve is a measure of the overall performance of the binary classification model, where AUC is the area under the ROC curve. A higher AUC value indicates that the classifier is able to differentiate between positive and negative instances better. A perfect classifier would have an AUC of 1. In general, the higher the AUC, the better the classifier. According to the result above, Gradient Boosting Decision Tree model after hyperparameter tuned has the highest AUC, and hence it is the model with the best performance among all the models here.

## 5.2 Testing Setup and Result

Based on the result of the performance evaluation, it is concluded that Gradient Boosting Decision Tree model has the best performance among the three models. Hence, Gradient Boosting Decision Tree model will be implemented into the GUI which is created to predict whether borrower has defaulted on loan or not. Figure below shows the GUI created and sample result of prediction.

The screenshot shows a web application titled "Credit Risk Prediction". The form is organized into three columns. The first column contains fields for NAME, LIMIT BALANCE, Sex (1-Male/2-Female), Education, Marriage, Age, and six Repayment status fields. The second column contains six BILL AMOUNT fields and six PAY AMOUNT fields. The third column contains a "Notes" section with detailed instructions for each input field. At the bottom, there are "Reset" and "Predict" buttons.

**Credit Risk Prediction**

NAME:  BILL AMOUNT 1:  Notes: For education, input 0 for others, 1 for graduate school, 2 for university, 3 for high school.

LIMIT BALANCE:  BILL AMOUNT 2:  For marriage, input 0 for others, 1 for married, 2 for single.

Sex (1-Male/2-Female):  BILL AMOUNT 3:  For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month,

Education:  BILL AMOUNT 4:  3 for payment delay 3 month, 4 for payment delay 4 month,

Marriage:  BILL AMOUNT 5:  5 for payment delay 5 month, 6 for payment delay 6 month,

Age:  BILL AMOUNT 6:  7 for payment delay 7 month, 8 for payment delay 8 month,

Repayment status 1:  PAY AMOUNT 1:  9 for payment delay 9 months and above.

Repayment status 2:  PAY AMOUNT 2:

Repayment status 3:  PAY AMOUNT 3:

Repayment status 4:  PAY AMOUNT 4:

Repayment status 5:  PAY AMOUNT 5:

Repayment status 6:  PAY AMOUNT 6:

Figure 5. 22: GUI created for credit risk prediction

If the form is not filled properly or there is blank field in the form, the system will show an error message when the user clicked the predict button. For example, the "NAME" field in the figure below is not filled, a message of "Name field is Empty!!" is shown when the user clicked the predict button.

This screenshot is identical to Figure 5.22, but with an additional yellow error message box at the bottom right that reads "Name Field is Empty!!". The "NAME" field in the first column is currently empty.

**Credit Risk Prediction**

NAME:  BILL AMOUNT 1:  Notes: For education, input 0 for others, 1 for graduate school, 2 for university, 3 for high school.

LIMIT BALANCE:  BILL AMOUNT 2:  For marriage, input 0 for others, 1 for married, 2 for single.

Sex (1-Male/2-Female):  BILL AMOUNT 3:  For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month,

Education:  BILL AMOUNT 4:  3 for payment delay 3 month, 4 for payment delay 4 month,

Marriage:  BILL AMOUNT 5:  5 for payment delay 5 month, 6 for payment delay 6 month,

Age:  BILL AMOUNT 6:  7 for payment delay 7 month, 8 for payment delay 8 month,

Repayment status 1:  PAY AMOUNT 1:  9 for payment delay 9 months and above.

Repayment status 2:  PAY AMOUNT 2:

Repayment status 3:  PAY AMOUNT 3:

Repayment status 4:  PAY AMOUNT 4:

Repayment status 5:  PAY AMOUNT 5:

Repayment status 6:  PAY AMOUNT 6:

Name Field is Empty!!

Figure 5. 23: System will show error message if there is blank field found

If the form is filled in properly or there is no blank field in the form, the system will show the result of the prediction when the user clicked the predict button. For example, in the figure below, you can see the result of the prediction at the bottom

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

## Chapter 5 System Evaluation and Discussion

right of the GUI when the user clicked the predict button. Besides that, the information of the borrower and the result of the prediction will be saved in a text file with the name of the borrower.

Figure 5. 24: System will display the result at the bottom right

A series of testing was carried out in order to ensure the accuracy of the credit risk prediction. The table below indicates the number of correct results of credit risk prediction. The details are filled out based on the details of the dataset and the result is compared.

Number of tests	Prediction Result	Correct/Wrong
1	Not defaulted on loan	Correct
2	Not defaulted on loan	Correct
3	Has defaulted on loan	Correct
4	Not defaulted on loan	Correct
5	Has defaulted on loan	Wrong
6	Has defaulted on loan	Correct
7	Has defaulted on loan	Wrong
8	Not defaulted on loan	Correct
9	Has defaulted on loan	Correct
10	Not defaulted on loan	Correct

Table 5. 1 Result of Prediction

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR



## Test 1:

The screenshot shows a web application titled "Credit Risk Prediction". It contains a form with various input fields for user data and loan details. The data entered for user "hee" is as follows:

Field	Value
NAME	hee
LIMIT BALANCE	90000
Sex (1-Male/2-Female)	2
Education	2
Marriage	2
Age	34
Repayment status 1	0
Repayment status 2	0
Repayment status 3	0
Repayment status 4	0
Repayment status 5	0
Repayment status 6	0
BILL AMOUNT 1	29239
BILL AMOUNT 2	14027
BILL AMOUNT 3	13559
BILL AMOUNT 4	14331
BILL AMOUNT 5	14948
BILL AMOUNT 6	15549
PAY AMOUNT 1	1518
PAY AMOUNT 2	1500
PAY AMOUNT 3	1000
PAY AMOUNT 4	1000
PAY AMOUNT 5	1000
PAY AMOUNT 6	5000

At the bottom, there are "Reset" and "Predict" buttons. The "Predict" button has been clicked, resulting in a green message box that says "hee not defaulted on loan!!!" and a yellow message box that says "hee's credit risk prediction report is saved in hee.txt".

Figure 5. 25: Test 1

Number of trials: 10

Number of correct classification result: 8

Number of wrong classification result: 2

Correct rate:  $8/10 \times 100\% = 80\%$

Wrong rate:  $2/10 \times 100\% = 20\%$

Based on the statistics above, the outcome of the prediction is considered highly accurate as it only failed 2 times out of 10 tests. In other words, the wrong rate is 20% out of 100% and the correct rate is 80% out of 100%. This means that the model is working well, and the GUI is also working well for prediction.

### 5.3 Project Challenges

There were several challenges that faced in this project. The following are the challenges in this project.

- During the implementation phase, I was faced with a problem when hyperparameter tuning the SVM (Linear kernel & RBF kernel) models. Due to low specification of laptop, the running time of hyperparameter tuning code is longer than expected. Large number of samples (30000) cause it to take a long time to get the results, up to a day. SVM models are known to have a

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

relatively high computational cost compared to other machine learning algorithms, especially when working with large datasets so it will take up more memory and more time for the mathematical calculation hence increasing the training time. To solve this problem, I have to reduce the number of samples by random sampling.

- Since I'm new to machine learning field and only learned basic things about machine learning before, I have limited and minimal knowledge about machine learning, the whole process of developing the models which include the feature engineering, hyperparameter tuning, etc. requires longer time to explore and make it works. Three models are developed and compared to get the best model.
- Developing GUI on Jupyter Notebook with the help of Tkinter library is a new thing for me. It took some time for me to explore and make the implementation of the model into the GUI works as expected.

### **5.3 Objectives Evaluation**

The first objective of this project is to develop a few robust binary classifiers for credit risk assessment. The main objective of this project is to develop Logistic Regression, gradient boosting decision tree and Support Vector Machine models for credit risk prediction. This objective is successfully achieved as the three models are successfully developed. These three models are compared and assessed at the end of the modelling process using a variety of performance metrics such as the confusion matrix, AUC-ROC curve, F1 score, accuracy, precision and recall. All these three models are able to predict whether the borrower has defaulted on loan or not with over 60% accuracy. In fact, the Gradient Boosting Decision Tree shows a high accuracy of 86.34% after hyperparameter tuned and is chosen to be the best model.

The second objective of this project is to develop a graphical user interface (GUI) for the credit risk prediction using Tkinter in python. After comparing the three models, the model with the best performance will be chosen and deployed into the GUI to predict the credit risk using real data. The Gradient Boosting Decision Tree is chosen to be implemented into the GUI to predict whether the borrower has defaulted on the loan or not. This objective is successfully achieved as the GUI was successfully created and works as expected. This can be proven in the System Testing session of this report.

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

### 5.3 Concluding Remark

The system testing was carried out and performance metrics were obtained to prove the accuracy of the models. From the testing results, it shown that the accuracy of the chosen model to implement into the GUI which is the Gradient Boosting Decision Tree model is relatively high. It is also proven that the GUI created works well and is able to make predictions. The project challenges were also identified. Finally, it can be said that all the objectives were achieved. Hence, the final outcome of this project is quite successful.

	ACCURACY	PRECISION	RECALL	F1 SCORE	AUC
LR	0.7037	0.7708	0.5813	0.6628	0.70
SVM (Linear)	0.7825	0.4816	0.5219	0.5010	0.69
SVM (RBF)	0.8225	0.6980	0.2670	0.3862	0.62
GDBT	0.8635	0.9117	0.8053	0.8553	0.86

Table 5. 2: Summary of Evaluation

## Chapter 6: Conclusion and Recommendation

### 6.1 Conclusion

Credit risk assessment is significant for the financial industry, especially the banking industry nowadays. Before the widespread of credit risk assessment using statistical model, the loan applications are reviewed and approved by human. However, the result of loan applications will take a long time to be produced. This is because manpower is limited. The workers have to review all the information of each applicant in order to decide whether approve or reject the application. This problem can be solved using the credit risk model.

For this project, Logistic Regression, Support Vector Machine (Linear kernel and RBF kernel) and Gradient Boosting Decision Tree models are developed. The methods that were used to develop the models are data acquisition, data exploration and visualization, data preparation, model training and performance evaluation. The main goal of this project is to develop a few robust credit risk models. In data acquisition, dataset was downloaded from a public repository platform called Kaggle. This dataset is used to develop the models to further validate the robustness of the model. Data exploration and data visualization is done to understand the structure and the information of the datasets. The data types of the attributes, data frame of the attributes, etc. are explored. The presence of missing value is checked. Data visualisation is also done to further understand the data. It helps to understand the relationship with the attributes. After that, data is prepared before the model training. The missing value is filled, the value of numerical categorical features is converted into categorical values and other necessary preparation is done. After the data preparation, the model is trained. The data is splitted into test set and train set. The train set is used to train the model and test set is used to validate the model. After that, the performance of the models is evaluated using several performance metrics which are accuracy, precision, recall and f1 score. The accuracy of the models is relatively low and still can be improved. Hence, hyperparameter tuning was done for the Support Vector Machine model and Gradient Boosting Decision Tree. Obviously, the accuracy of the models got boosted. The models are compared and the model with the best performance is the Gradient Boosting Decision Tree model. This model is used to implement into the graphical user interface (GUI) created which is the second objective of this project. The second objective of this project is to develop a GUI for Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

credit risk prediction. At the end of the project, the project objectives are achieved. Through the system testing, this project also can be considered relatively reliable. Provided that the GUI is able to make the prediction with the Gradient Boosting Decision Tree model and the result is pretty accurate.

## **6.2 Recommendation**

Enhance the created GUI to include probability estimates for credit risk assessment. Probability estimates provide valuable information about the likelihood of default and help quantify the level of risk associated with each borrower. Including this feature in the GUI will enable users to make more informed decisions and assess the uncertainty associated with each credit assessment. Besides that, expand the GUI to incorporate a module for predicting the amount that borrowers are likely to pay in the upcoming month. Utilize historical payment data, borrower characteristics, and relevant factors to build a predictive model specifically for estimating next month's payment amounts. This will provide users with a more comprehensive understanding of borrowers' repayment capacity.

## REFERENCE

- [1] R. Cole -Federal Reserve Board *et al.*, “Principles for the Management of Credit Risk Basel Committee on Banking Supervision Basel September 2000 Risk Management Group of the Basel Committee on Banking Supervision Chairman: Commission Bancaire et Financière, Brussels Mr Jos Meuleman Office of the Superintendent of Financial Institutions, Ottawa Ms Aina Liepins Secretariat of the Basel Committee on Banking Supervision, Bank for International Settlements.”
- [2] Badreesh Shetty, “An in-depth guide to supervised machine learning classification,” <https://builtin.com/data-science/supervised-machine-learning-classification>, Jul. 17, 2019.
- [3] Avijeet Biswal, “Top 10 Deep Learning Algorithms You Should Know in 2022,” <https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm>, Jul. 19, 2022.
- [4] JavaTPoint, “Installing Anaconda and Python,” <https://www.javatpoint.com/machine-learning-installing-anaconda-and-python>.
- [5] Vijay Kanade, “Top 10 Python Machine Learning Libraries in 2022,” <https://www.spiceworks.com/tech/artificial-intelligence/articles/top-python-machine-learning-libraries/>, May 06, 2022.
- [6] GeeksforGeeks, “Best Python libraries for Machine Learning,” <https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/>, Aug. 22, 2022.
- [7] Joblib developers, “Joblib: running Python functions as pipeline jobs,” <https://joblib.readthedocs.io/en/latest/>, 2021.
- [8] tutorialspoint, “Python - GUI Programming (Tkinter),” [https://www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm), 2023.
- [9] python, “What is Python? Executive Summary,” <https://www.python.org/doc/essays/blurb/>, 2023.
- [10] Z. Tian, J. Xiao, H. Feng, and Y. Wei, “Credit Risk Assessment based on Gradient Boosting Decision Tree,” in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 150–160. doi: 10.1016/j.procs.2020.06.070.
- [11] B. Okemwa, “Consumer credit risk modelling using machine learning algorithms: a comparative approach.” [Online]. Available: <https://su-plus.strathmore.edu/handle/11071/6789><http://su-plus.strathmore.edu/handle/11071/6789>
- [12] CFI Team, “Correlation Matrix,” <https://corporatefinanceinstitute.com/resources/excel/study/correlation-matrix/>, Mar. 05, 2022.

## Reference

- [13] Jason Brownlee, “SMOTE for Imbalanced Classification with Python,” <https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>, Mar. 17, 2021.
- [14] “Machine learning in credit risk: Evaluation of supervised machine learning models predicting credit risk in the financial sector.”
- [15] Sunil Ray, “Understanding Support Vector Machine(SVM) algorithm from examples (along with code),” <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>, Aug. 26, 2021.
- [16] IBM, “How SVM Works,” <https://www.ibm.com/docs/it/spss-modeler/saas?topic=models-how-svm-works>, Aug. 17, 2021.
- [17] Gaurav, “An Introduction to Gradient Boosting Decision Trees,” <https://www.machinelearningplus.com/machine-learning/an-introduction-to-gradient-boosting-decision-trees/>, Jun. 12, 2021.
- [18] scikit-learn developers, “3.2. Tuning the hyper-parameters of an estimator,” [https://scikit-learn.org/stable/modules/grid\\_search.html](https://scikit-learn.org/stable/modules/grid_search.html), 2023.
- [19] scikit-learn developers, “RBF SVM parameters,” [https://scikit-learn.org/stable/auto\\_examples/svm/plot\\_rbf\\_parameters.html](https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html), 2023.
- [20] xgboost developers, “XGBoost Parameters,” <https://xgboost.readthedocs.io/en/stable/parameter.html>, 2022.
- [21] Jason Brownlee, “What is a Confusion Matrix in Machine Learning,” <https://machinelearningmastery.com/confusion-matrix-machine-learning/>, Nov. 18, 2016.
- [22] Mohammed Sunasra, “Performance Metrics for Classification problems in Machine Learning,” <https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>, Nov. 12, 2017.

# APPENDIX

## TEST RESULT OF PREDICTION WITH GUI

Test 2:

**Credit Risk Prediction**

Credit Risk Prediciton

NAME:	hee	BILL AMOUNT 1:	11285	<b>Notes:</b> For education, input 0 for others, 1 for graduate school, 2 for university, 3 for high school. For marriage, input 0 for others, 1 for married, 2 for single. For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.
LIMIT BALANCE:	140000	BILL AMOUNT 2:	14096	
Sex (1-Male/2-Female):	2	BILL AMOUNT 3:	12108	
Education:	1	BILL AMOUNT 4:	12211	
Marriage:	3	BILL AMOUNT 5:	11793	
Age:	28	BILL AMOUNT 6:	3719	
Repayment status 1:	0	PAY AMOUNT 1:	3329	
Repayment status 2:	0	PAY AMOUNT 2:	0	
Repayment status 3:	2	PAY AMOUNT 3:	432	
Repayment status 4:	0	PAY AMOUNT 4:	1000	
Repayment status 5:	0	PAY AMOUNT 5:	1000	
Repayment status 6:	0	PAY AMOUNT 6:	1000	

Reset Predict

**hee not defaulted on loan!!!**  
 hee's credit risk prediction report is saved in hee.txt

Test 3:

**Credit Risk Prediction**

Credit Risk Prediciton

NAME:	hee	BILL AMOUNT 1:	-109	<b>Notes:</b> For education, input 0 for others, 1 for graduate school, 2 for university, 3 for high school. For marriage, input 0 for others, 1 for married, 2 for single. For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.
LIMIT BALANCE:	60000	BILL AMOUNT 2:	-425	
Sex (1-Male/2-Female):	1	BILL AMOUNT 3:	259	
Education:	2	BILL AMOUNT 4:	-57	
Marriage:	1	BILL AMOUNT 5:	127	
Age:	27	BILL AMOUNT 6:	-189	
Repayment status 1:	1	PAY AMOUNT 1:	0	
Repayment status 2:	-2	PAY AMOUNT 2:	1000	
Repayment status 3:	-1	PAY AMOUNT 3:	0	
Repayment status 4:	-1	PAY AMOUNT 4:	500	
Repayment status 5:	-1	PAY AMOUNT 5:	0	
Repayment status 6:	-1	PAY AMOUNT 6:	1000	

Reset Predict

**hee has defaulted on loan.**  
 hee's credit risk prediction report is saved in hee.txt



## Appendix

### Test 4:

**Credit Risk Prediction**

**Credit Risk Prediciton**

NAME:	hee	BILL AMOUNT 1:	93036	<b>Notes:</b> For education, input 0 for others, 1 for graduate school, 2 for university ,3 for high school. For marriage, input 0 for others, 1 for married, 2 for single. For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.
LIMIT BALANCE:	10000	BILL AMOUNT 2:	84071	
Sex (1-Male/2-Female):	1	BILL AMOUNT 3:	82880	
Education:	1	BILL AMOUNT 4:	80958	
Marriage:	2	BILL AMOUNT 5:	78703	
Age:	32	BILL AMOUNT 6:	75589	
Repayment status 1:	0	PAY AMOUNT 1:	3023	
Repayment status 2:	0	PAY AMOUNT 2:	3511	
Repayment status 3:	0	PAY AMOUNT 3:	3302	
Repayment status 4:	0	PAY AMOUNT 4:	3204	
Repayment status 5:	0	PAY AMOUNT 5:	3200	
Repayment status 6:	0	PAY AMOUNT 6:	2504	

Reset Predict

**hee not defaulted on loan!!!**

hee's credit risk prediction report is saved in hee.txt

### Test 5:

**Credit Risk Prediction**

**Credit Risk Prediciton**

NAME:	hee	BILL AMOUNT 1:	30265	<b>Notes:</b> For education, input 0 for others, 1 for graduate school, 2 for university ,3 for high school. For marriage, input 0 for others, 1 for married, 2 for single. For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.
LIMIT BALANCE:	160000	BILL AMOUNT 2:	-131	
Sex (1-Male/2-Female):	1	BILL AMOUNT 3:	-527	
Education:	1	BILL AMOUNT 4:	-923	
Marriage:	2	BILL AMOUNT 5:	-1488	
Age:	30	BILL AMOUNT 6:	-1884	
Repayment status 1:	-1	PAY AMOUNT 1:	131	
Repayment status 2:	-1	PAY AMOUNT 2:	396	
Repayment status 3:	-2	PAY AMOUNT 3:	396	
Repayment status 4:	-2	PAY AMOUNT 4:	565	
Repayment status 5:	-2	PAY AMOUNT 5:	792	
Repayment status 6:	-1	PAY AMOUNT 6:	4	

Reset Predict

**hee has defaulted on loan.!!**

hee's credit risk prediction report is saved in hee.txt

## Appendix

### Test 6:

Credit Risk Prediction

Credit Risk Prediction

NAME: hee
LIMIT BALANCE: 150000
Sex (1-Male/2-Female): 2
Education: 5
Marriage: 2
Age: 46
Repayment status 1: 0
Repayment status 2: 0
Repayment status 3: -1
Repayment status 4: 0
Repayment status 5: 0
Repayment status 6: -2

BILL AMOUNT 1: 4463
BILL AMOUNT 2: 3034
BILL AMOUNT 3: 1170
BILL AMOUNT 4: 1170
BILL AMOUNT 5: 0
BILL AMOUNT 6: 0
PAY AMOUNT 1: 1013
PAY AMOUNT 2: 1170
PAY AMOUNT 3: 0
PAY AMOUNT 4: 0
PAY AMOUNT 5: 0
PAY AMOUNT 6: 0

Notes:  
For education, input 0 for others, 1 for graduate school, 2 for university, 3 for high school.  
For marriage, input 0 for others, 1 for married, 2 for single.  
For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.

Reset
Predict

hee has defaulted on loan.  
hee's credit risk prediction report is saved in hee.txt

### Test 7:

Credit Risk Prediction

Credit Risk Prediction

NAME: hee
LIMIT BALANCE: 380000
Sex (1-Male/2-Female): 2
Education: 1
Marriage: 2
Age: 30
Repayment status 1: -2
Repayment status 2: -2
Repayment status 3: -1
Repayment status 4: 0
Repayment status 5: 0
Repayment status 6: 0

BILL AMOUNT 1: -81
BILL AMOUNT 2: -303
BILL AMOUNT 3: 32475
BILL AMOUNT 4: 32891
BILL AMOUNT 5: 33564
BILL AMOUNT 6: 34056
PAY AMOUNT 1: 223
PAY AMOUNT 2: 33178
PAY AMOUNT 3: 1171
PAY AMOUNT 4: 1197
PAY AMOUNT 5: 1250
PAY AMOUNT 6: 5000

Notes:  
For education, input 0 for others, 1 for graduate school, 2 for university, 3 for high school.  
For marriage, input 0 for others, 1 for married, 2 for single.  
For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.

Reset
Predict

hee has defaulted on loan.  
hee's credit risk prediction report is saved in hee.txt

## Appendix

### Test 8:

Credit Risk Prediction

### Credit Risk Prediciton

NAME:	hee	BILL AMOUNT 1:	115785	<b>Notes:</b> For education, input 0 for others. 1 for graduate school, 2 for university, 3 for high school. For marriage, input 0 for others. 1 for married, 2 for single. For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.
LIMIT BALANCE:	210000	BILL AMOUNT 2:	122904	
Sex (1-Male/2-Female):	1	BILL AMOUNT 3:	129847	
Education:	3	BILL AMOUNT 4:	137277	
Marriage:	1	BILL AMOUNT 5:	145533	
Age:	45	BILL AMOUNT 6:	154105	
Repayment status 1:	2	PAY AMOUNT 1:	10478	
Repayment status 2:	3	PAY AMOUNT 2:	10478	
Repayment status 3:	4	PAY AMOUNT 3:	11078	
Repayment status 4:	4	PAY AMOUNT 4:	11078	
Repayment status 5:	5	PAY AMOUNT 5:	11678	
Repayment status 6:	6	PAY AMOUNT 6:	10478	

Reset Predict

**hee not defaulted on loan!!!**  
 hee's credit risk prediction report is saved in hee.txt

### Test 9:

Credit Risk Prediction

### Credit Risk Prediciton

NAME:	hee	BILL AMOUNT 1:	47790	<b>Notes:</b> For education, input 0 for others. 1 for graduate school, 2 for university, 3 for high school. For marriage, input 0 for others. 1 for married, 2 for single. For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.
LIMIT BALANCE:	50000	BILL AMOUNT 2:	18114	
Sex (1-Male/2-Female):	1	BILL AMOUNT 3:	18250	
Education:	2	BILL AMOUNT 4:	-14	
Marriage:	1	BILL AMOUNT 5:	72	
Age:	36	BILL AMOUNT 6:	658	
Repayment status 1:	0	PAY AMOUNT 1:	2000	
Repayment status 2:	0	PAY AMOUNT 2:	1000	
Repayment status 3:	0	PAY AMOUNT 3:	2000	
Repayment status 4:	0	PAY AMOUNT 4:	500	
Repayment status 5:	-1	PAY AMOUNT 5:	1000	
Repayment status 6:	-1	PAY AMOUNT 6:	20011	

Reset Predict

**hee has defaulted on loan.**  
 hee's credit risk prediction report is saved in hee.txt

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

## Appendix Test 10:

Credit Risk Prediction

### Credit Risk Prediciton

NAME:	hee	BILL AMOUNT 1:	46140	Notes:
LIMIT BALANCE:	330000	BILL AMOUNT 2:	45781	For education, input 0 for others, 1 for graduate school, 2 for university, 3 for high school.
Sex (1-Male/2-Female):	1	BILL AMOUNT 3:	48139	For marriage, input 0 for others, 1 for married, 2 for single.
Education:	1	BILL AMOUNT 4:	51137	For Repayment status, input -1 for pay duly, 0 for no delay, 1 for payment delay 1 month, 2 for payment delay 2 month, 3 for payment delay 3 month, 4 for payment delay 4 month, 5 for payment delay 5 month, 6 for payment delay 6 month, 7 for payment delay 7 month, 8 for payment delay 8 month, 9 for payment delay 9 months and above.
Marriage:	2	BILL AMOUNT 5:	39450	
Age:	25	BILL AMOUNT 6:	25358	
Repayment status 1:	0	PAY AMOUNT 1:	2504	
Repayment status 2:	0	PAY AMOUNT 2:	4007	
Repayment status 3:	0	PAY AMOUNT 3:	5056	
Repayment status 4:	0	PAY AMOUNT 4:	74	
Repayment status 5:	2	PAY AMOUNT 5:	1023	
Repayment status 6:	0	PAY AMOUNT 6:	2564	

Reset Predict

hee not defaulted on loan!!!

hee's credit risk prediction report is saved in hee.txt

**CODE OF GRAPHICAL USER INTERFACE (GUI)**

```

In [1]: import re
        from tkinter import *
        import tkinter as tk
        from tkinter import filedialog
        from tkinter import messagebox

        import joblib
        import skimage
        from skimage.io import imread
        from skimage.transform import resize
        import numpy as np

In [4]: from PIL import ImageTk, Image
        from tkinter import ttk
        import pandas as pd

        global DB
        DF = pd.DataFrame()
        clf = joblib.load('model1.pkl')
        global a

        def filedreq():

            if name.get() == "":
                print("Name Field is Empty!!")
                user = "Name Field is Empty!!"
                Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

            elif limit_bal.get() == "":
                print("Limit Balance Field is Empty!!")
                user = "Limit Balance Field is Empty!!"
                Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

            elif sex.get() == "":
                print("Sex Field is Empty!!")
                user = "Sex Field is Empty!!"
                Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

            elif education.get() == "":
                print("Education Field is Empty!!")
                user = "Education Field is Empty!!"
                Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

            elif marriage.get() == "":
                print("Marriage Field is Empty!!")
                user = "Marriage Field is Empty!!"
                Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

```

## Appendix

```
— elif age.get() == "":
—     print("Age Field is Empty!!")
—     user = "Age Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

— elif pay_0.get() == "":
—     print("Repayment status 1 Field is Empty!!")
—     user = "Repayment status 1 Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

— elif pay_2.get() == "":
—     print("Repayment status 2 Field is Empty!!")
—     user = "Repayment status 2 Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

— elif pay_3.get() == "":
—     print("Repayment status 3 Field is Empty!!")
—     user = "Repayment status 3 Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

— elif pay_4.get() == "":
—     print("Repayment status 4 Field is Empty!!")
—     user = "Repayment status 4 Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

— elif pay_5.get() == "":
—     print("Repayment status 5 Field is Empty!!")
—     user = "Repayment status 5 Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

— elif pay_6.get() == "":
—     print("Repayment status 6 Field is Empty!!")
—     user = "Repayment status 6 Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

— elif bill_amt1.get() == "":
—     print("BILL AMOUNT 1 Field is Empty!!")
—     user = "BILL AMOUNT 1 Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

— elif bill_amt2.get() == "":
—     print("BILL AMOUNT 2 Field is Empty!!")
—     user = "BILL AMOUNT 2 Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

— elif bill_amt3.get() == "":
—     print("BILL AMOUNT 3 Field is Empty!!")
—     user = "BILL AMOUNT 3 Field is Empty!!"
—     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)
```

```

elif bill_amt4.get() == "":
    print("BILL AMOUNT 4 Field is Empty!!")
    user = "BILL AMOUNT 4 Field is Empty!!"
    Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

elif bill_amt5.get() == "":
    print("BILL AMOUNT 5 Field is Empty!!")
    user = "BILL AMOUNT 5 Field is Empty!!"
    Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

elif bill_amt6.get() == "":
    print("BILL AMOUNT 6 Field is Empty!!")
    user = "BILL AMOUNT 6 Field is Empty!!"
    Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

elif pay_amt1.get() == "":
    print("PAY AMOUNT 1 Field is Empty!!")
    user = "PAY AMOUNT 1 Field is Empty!!"
    Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

elif pay_amt2.get() == "":
    print("PAY AMOUNT 2 Field is Empty!!")
    user = "PAY AMOUNT 2 Field is Empty!!"
    Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

elif pay_amt3.get() == "":
    print("PAY AMOUNT 3 Field is Empty!!")
    user = "PAY AMOUNT 3 Field is Empty!!"
    Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

elif pay_amt4.get() == "":
    print("PAY AMOUNT 4 Field is Empty!!")
    user = "PAY AMOUNT 4 Field is Empty!!"
    Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

elif pay_amt5.get() == "":
    print("PAY AMOUNT 5 Field is Empty!!")
    user = "PAY AMOUNT 5 Field is Empty!!"
    Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

elif pay_amt6.get() == "":
    print("PAY AMOUNT 6 Field is Empty!!")
    user = "PAY AMOUNT 6 Field is Empty!!"
    Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=630,y=500)

else:
    test()

```

## Appendix

```
def test():
    print("Testing...")
    # Test code will go here...
    DF = pd.DataFrame(columns=['LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3', 'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6'])
    LIMIT_BAL=limit_bal.get()
    DF.loc[0, 'LIMIT_BAL'] = LIMIT_BAL
    SEX=sex.get()
    DF.loc[0, 'SEX'] = SEX
    EDUCATION=education.get()
    DF.loc[0, 'EDUCATION'] = EDUCATION
    MARRIAGE=marriage.get()
    DF.loc[0, 'MARRIAGE'] = MARRIAGE
    AGE=age.get()
    DF.loc[0, 'AGE'] = AGE
    PAY_0=pay_0.get()
    DF.loc[0, 'PAY_0'] = PAY_0
    PAY_2=pay_2.get()
    DF.loc[0, 'PAY_2'] = PAY_2
    PAY_3=pay_3.get()
    DF.loc[0, 'PAY_3'] = PAY_3
    PAY_4=pay_4.get()
    DF.loc[0, 'PAY_4'] = PAY_4
    PAY_5=pay_5.get()
    DF.loc[0, 'PAY_5'] = PAY_5
    PAY_6=pay_6.get()
    DF.loc[0, 'PAY_6'] = PAY_6
    BILL_AMT1=bill_amt1.get()
    DF.loc[0, 'BILL_AMT1'] = BILL_AMT1
    BILL_AMT2=bill_amt2.get()
    DF.loc[0, 'BILL_AMT2'] = BILL_AMT2
    BILL_AMT3=bill_amt3.get()
    DF.loc[0, 'BILL_AMT3'] = BILL_AMT3
    BILL_AMT4=bill_amt4.get()
    DF.loc[0, 'BILL_AMT4'] = BILL_AMT4
    BILL_AMT5=bill_amt5.get()
    DF.loc[0, 'BILL_AMT5'] = BILL_AMT5
    BILL_AMT6=bill_amt6.get()
    DF.loc[0, 'BILL_AMT6'] = BILL_AMT6
    PAY_AMT1=pay_amt1.get()
    DF.loc[0, 'PAY_AMT1'] = PAY_AMT1
    PAY_AMT2=pay_amt2.get()
    DF.loc[0, 'PAY_AMT2'] = PAY_AMT2
    PAY_AMT3=pay_amt3.get()
    DF.loc[0, 'PAY_AMT3'] = PAY_AMT3
    PAY_AMT4=pay_amt4.get()
    DF.loc[0, 'PAY_AMT4'] = PAY_AMT4
    PAY_AMT5=pay_amt5.get()
    DF.loc[0, 'PAY_AMT5'] = PAY_AMT5
    PAY_AMT6=pay_amt6.get()
    DF.loc[0, 'PAY_AMT6'] = PAY_AMT6
```



## Appendix

```
—*print(DF.shape)
—*DB=DF

—*result = clf.predict(DB)
—*# print(result)
—*if result[0] == 0:
—*     print("No")
—*     person = name.get()
—*     user = person + ' not defaulted on loan!!!'
—*     a = user
—*     Label(win,text=user,fg="red",bg="white",font = ("Calibri 12 bold")).place(x=600,y=500)
—*else:
—*     print("Yes")
—*     person = name.get()
—*     user = person + ' has defaulted on loan.'
—*     a = user
—*     Label(win,text=user,fg="blue",bg="yellow",font = ("Calibri 12 bold")).place(x=600,y=500)

—*# After Test, save() will execute
—*save(a)

def save(a):

—*NAME = name.get()
—*limitBal = limit_bal.get()
—*SEX = sex.get()
—*MARRIAGE = marriage.get()
—*EDUCATION = education.get()
—*AGE = age.get()
—*pay0 = pay_0.get()
—*pay2 = pay_2.get()
—*pay3 = pay_3.get()
—*pay4 = pay_4.get()
—*pay5 = pay_5.get()
—*pay6 = pay_6.get()
—*billAmt1 = bill_amt1.get()
—*billAmt2 = bill_amt2.get()
—*billAmt3 = bill_amt3.get()
—*billAmt4 = bill_amt4.get()
—*billAmt5 = bill_amt5.get()
—*billAmt6 = bill_amt6.get()
—*payAmt1 = pay_amt1.get()
—*payAmt2 = pay_amt2.get()
—*payAmt3 = pay_amt3.get()
—*payAmt4 = pay_amt4.get()
—*payAmt5 = pay_amt5.get()
—*payAmt6 = pay_amt6.get()
—*save_name = NAME+".txt"
```

## Appendix

```
—*
—*file = open(save_name,"a")
—*file.write("\n\nName: "+NAME+"\n")
—*file.write("\n\nLimit Balance: "+limitBal+"\n")
—*file.write("Sex: "+SEX+"\n")
—*file.write("Marriage: "+MARRIAGE+"\n")
—*file.write("Education: "+EDUCATION+"\n")
—*file.write("Age: "+AGE+"\n")
—*file.write("PAY_0: "+pay0+"\n")
—*file.write("PAY_2: "+pay2+"\n")
—*file.write("PAY_3: "+pay3+"\n")
—*file.write("PAY_4: "+pay4+"\n")
—*file.write("PAY_5: "+pay5+"\n")
—*file.write("PAY_6: "+pay6+"\n")
—*file.write("BILL AMT1: "+billAnt1+"\n")
—*file.write("BILL AMT2: "+billAnt2+"\n")
—*file.write("BILL AMT3: "+billAnt3+"\n")
—*file.write("BILL AMT4: "+billAnt4+"\n")
—*file.write("BILL AMT5: "+billAnt5+"\n")
—*file.write("BILL AMT6: "+billAnt6+"\n")
—*file.write("PAY AMT1: "+payAnt1+"\n")
—*file.write("PAY AMT2: "+payAnt2+"\n")
—*file.write("PAY AMT3: "+payAnt3+"\n")
—*file.write("PAY AMT4: "+payAnt4+"\n")
—*file.write("PAY AMT5: "+payAnt5+"\n")
—*file.write("PAY AMT6: "+payAnt6+"\n")
—*file.write("Report: "+ a +"\n")
—*file.close()
—report = NAME + "'s credit risk prediction report is saved in "+NAME+".txt"
—Label(win,text=report,fg="blue",bg="yellow",font = ("Calibri 10 bold")).place(x=600,y=530)
—# print("Printing Data: ")
—# print(First,Last,phone,email,address,gender)

def reset():
—name.set("")
—limit_bal.set("")
—sex.set("")
—education.set("")
—marriage.set("")
—age.set("")
—pay_0.set("")
—pay_2.set("")
—pay_3.set("")
—pay_4.set("")
—pay_5.set("")
—pay_6.set("")
—bill_ant1.set("")
—bill_ant2.set("")
—bill_ant3.set("")
—bill_ant4.set("")
—bill_ant5.set("")
```

## Appendix

```
bill_amt6.set("")
pay_amt1.set("")
pay_amt2.set("")
pay_amt3.set("")
pay_amt4.set("")
pay_amt5.set("")
pay_amt6.set("")

win = Tk()

win.geometry("1200x600")
win.configure(background="cyan")
win.resizable(False, False)
win.title("Credit Risk Prediction")
#win.iconbitmap('icon.ico')

title = Label(win, text="Credit Risk Prediciton", bg="gray", width="300", height="2", fg="White", font = ("Calibri 20 bold italic unde

name = Label(win, text="NAME: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=100)
gap = Label(win, text="", bg="cyan").pack()

limit_bal = Label(win, text="LIMIT BALANCE: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=130)
gap = Label(win, text="", bg="cyan").pack()

sex = Label(win, text="Sex (1-Male/2-Female): ", bg="cyan", font = ("Verdana 10")).place(x=12, y=160)
gap = Label(win, text="", bg="cyan").pack()

education = Label(win, text="Education: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=190)
gap = Label(win, text="", bg="cyan").pack()

marriage = Label(win, text="Marriage: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=220)
gap = Label(win, text="", bg="cyan").pack()

age = Label(win, text="Age: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=250)
gap = Label(win, text="", bg="cyan").pack()

pay_0 = Label(win, text="Repayment status 1: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=280)
gap = Label(win, text="", bg="cyan").pack()

pay_2 = Label(win, text="Repayment status 2: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=310)
gap = Label(win, text="", bg="cyan").pack()

pay_3 = Label(win, text="Repayment status 3: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=340)
gap = Label(win, text="", bg="cyan").pack()

pay_4 = Label(win, text="Repayment status 4: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=370)
gap = Label(win, text="", bg="cyan").pack()

pay_5 = Label(win, text="Repayment status 5: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=400)
```

## Appendix

```
gap = Label(win, text="", bg="cyan").pack()

pay_6 = Label(win, text="Repayment status 6: ", bg="cyan", font = ("Verdana 12")).place(x=12, y=430)
gap = Label(win, text="", bg="cyan").pack()

bill_amt1 = Label(win, text="BILL AMOUNT 1: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=100)
gap = Label(win, text="", bg="cyan").pack()

bill_amt2 = Label(win, text="BILL AMOUNT 2: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=130)
gap = Label(win, text="", bg="cyan").pack()

bill_amt3 = Label(win, text="BILL AMOUNT 3: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=160)
gap = Label(win, text="", bg="cyan").pack()

bill_amt4 = Label(win, text="BILL AMOUNT 4: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=190)
gap = Label(win, text="", bg="cyan").pack()

bill_amt5 = Label(win, text="BILL AMOUNT 5: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=220)
gap = Label(win, text="", bg="cyan").pack()

bill_amt6 = Label(win, text="BILL AMOUNT 6: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=250)
gap = Label(win, text="", bg="cyan").pack()

pay_amt1 = Label(win, text="PAY AMOUNT 1: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=280)
gap = Label(win, text="", bg="cyan").pack()

pay_amt2 = Label(win, text="PAY AMOUNT 2: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=310)
gap = Label(win, text="", bg="cyan").pack()

pay_amt3 = Label(win, text="PAY AMOUNT 3: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=340)
gap = Label(win, text="", bg="cyan").pack()

pay_amt4 = Label(win, text="PAY AMOUNT 4: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=370)
gap = Label(win, text="", bg="cyan").pack()

pay_amt5 = Label(win, text="PAY AMOUNT 5: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=400)
gap = Label(win, text="", bg="cyan").pack()

pay_amt6 = Label(win, text="PAY AMOUNT 6: ", bg="cyan", font = ("Verdana 12")).place(x=450, y=430)
gap = Label(win, text="", bg="cyan").pack()

text1 = Label(win, text="Notes: ", bg="cyan", font = ("Verdana 9")).place(x=820, y=100)
gap = Label(win, text="", bg="cyan").pack()

text2 = Label(win, text="For education, input 0 for others, 1 for graduate school, 2 for university", bg="cyan", font = ("Verdana 7"))
text3 = Label(win, text=", 3 for high school.", bg="cyan", font = ("Verdana 7")).place(x=820, y=140)
gap = Label(win, text="", bg="cyan").pack()

text4 = Label(win, text="For marriage, input 0 for others, 1 for married, 2 for single.", bg="cyan", font = ("Verdana 7"))
gap = Label(win, text="", bg="cyan").pack()
```

## Appendix

```
text5 = Label(win, text="For Repayment status, input -1 for pay duly, 0 for no delay,",bg="cyan",font = ("Verdana 7")).place(x=800,y=100)
text6 = Label(win, text="1 for payment delay 1 month, 2 for payment delay 2 month,",bg="cyan",font = ("Verdana 7")).place(x=820,y=110)
text7 = Label(win, text="3 for payment delay 3 month, 4 for payment delay 4 month,",bg="cyan",font = ("Verdana 7")).place(x=820,y=120)
text8 = Label(win, text="5 for payment delay 5 month, 6 for payment delay 6 month,",bg="cyan",font = ("Verdana 7")).place(x=820,y=130)
text9 = Label(win, text="7 for payment delay 7 month, 8 for payment delay 8 month,",bg="cyan",font = ("Verdana 7")).place(x=820,y=140)
text10 = Label(win, text="9 for payment delay 9 months and above.",bg="cyan",font = ("Verdana 7")).place(x=820,y=150)
gap = Label(win,text="",bg="cyan").pack()

name = StringVar()
limit_bal = StringVar()
sex = StringVar()
education = StringVar()
marriage = StringVar()
age = StringVar()
pay_0 = StringVar()
pay_2 = StringVar()
pay_3 = StringVar()
pay_4 = StringVar()
pay_5 = StringVar()
pay_6 = StringVar()
bill_amt1 = StringVar()
bill_amt2 = StringVar()
bill_amt3 = StringVar()
bill_amt4 = StringVar()
bill_amt5 = StringVar()
bill_amt6 = StringVar()
pay_amt1 = StringVar()
pay_amt2 = StringVar()
pay_amt3 = StringVar()
pay_amt4 = StringVar()
pay_amt5 = StringVar()
pay_amt6 = StringVar()

entry_name = Entry(win,textvariable = name,width=30)
entry_name.place(x=200,y=100)
entry_limitbal = Entry(win,textvariable = limit_bal,width=30)
entry_limitbal.place(x=200,y=130)
entry_sex = Entry(win,textvariable = sex,width=30)
entry_sex.place(x=200,y=160)
entry_education = Entry(win,textvariable = education,width=30)
entry_education.place(x=200,y=190)
entry_marriage = Entry(win,textvariable = marriage,width=30)
entry_marriage.place(x=200,y=220)
entry_age = Entry(win,textvariable = age,width=30)
entry_age.place(x=200,y=250)
entry_pay0 = Entry(win,textvariable = pay_0,width=30)
entry_pay0.place(x=200,y=280)
entry_pay2 = Entry(win,textvariable = pay_2,width=30)
entry_pay2.place(x=200,y=310)
```

## Appendix

```
entry_pay3 = Entry(win, textvariable = pay_3, width=30)
entry_pay3.place(x=200, y=340)
entry_pay4 = Entry(win, textvariable = pay_4, width=30)
entry_pay4.place(x=200, y=370)
entry_pay5 = Entry(win, textvariable = pay_5, width=30)
entry_pay5.place(x=200, y=400)
entry_pay6 = Entry(win, textvariable = pay_6, width=30)
entry_pay6.place(x=200, y=430)
entry_bill1 = Entry(win, textvariable = bill_amt1, width=30)
entry_bill1.place(x=630, y=100)
entry_bill2 = Entry(win, textvariable = bill_amt2, width=30)
entry_bill2.place(x=630, y=130)
entry_bill3 = Entry(win, textvariable = bill_amt3, width=30)
entry_bill3.place(x=630, y=160)
entry_bill4 = Entry(win, textvariable = bill_amt4, width=30)
entry_bill4.place(x=630, y=190)
entry_bill5 = Entry(win, textvariable = bill_amt5, width=30)
entry_bill5.place(x=630, y=220)
entry_bill6 = Entry(win, textvariable = bill_amt6, width=30)
entry_bill6.place(x=630, y=250)
entry_payamt1 = Entry(win, textvariable = pay_amt1, width=30)
entry_payamt1.place(x=630, y=280)
entry_payamt2 = Entry(win, textvariable = pay_amt2, width=30)
entry_payamt2.place(x=630, y=310)
entry_payamt3 = Entry(win, textvariable = pay_amt3, width=30)
entry_payamt3.place(x=630, y=340)
entry_payamt4 = Entry(win, textvariable = pay_amt4, width=30)
entry_payamt4.place(x=630, y=370)
entry_payamt5 = Entry(win, textvariable = pay_amt5, width=30)
entry_payamt5.place(x=630, y=400)
entry_payamt6 = Entry(win, textvariable = pay_amt6, width=30)
entry_payamt6.place(x=630, y=430)

path = Label(win, bg="cyan", font = ("Verdana 8"))
path.place(x=140, y=380)

reset = Button(win, text="Reset", width="12", height="1", activebackground="red", bg="Pink", font = ("Calibri 12 "), command = reset)
submit = Button(win, text="Predict", width="12", height="1", activebackground="violet", bg="Pink", command = filedreq, font = ("Cali

win.mainloop()
```

# WEEKLY REPORT

(Project II)

<b>Trimester, Year: Trimester 3, Year 3</b>	<b>Study week no.: 3</b>
<b>Student Name &amp; ID: Hee Tuck Hoe, 19ACB03841</b>	
<b>Supervisor: Dr. Lim Jia Qi</b>	
<b>Project Title: Predictive Modeling in credit risk assessment</b>	

## 1. WORK DONE

- Overviewed things done for Final Year Project 1.
- Discussed and made some changes on the project according to Moderator's suggestion.

## 2. WORK TO BE DONE

- Proceed developing rest of the models which are the Support Vector Machine (SVM) and Gradient Boosting Decision Tree models.

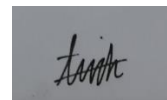
## 3. PROBLEMS ENCOUNTERED

## 4. SELF EVALUATION OF THE PROGRESS

- The progress is good.



Supervisor's signature



Student's signature

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

# WEEKLY REPORT

<b>Trimester, Year: Trimester 3, Year 3</b>	<b>Study week no.: 5</b>
<b>Student Name &amp; ID: Hee Tuck Hoe, 19ACB03841</b>	
<b>Supervisor: Dr. Lim Jia Qi</b>	
<b>Project Title: Predictive Modeling in credit risk assessment</b>	

## 1. WORK DONE

- Support Vector Machine (Linear and RBF kernel) model developed.
- Development of Gradient Boosting Decision Tree model almost completed.

## 2. WORK TO BE DONE

- Complete the Gradient Boosting Decision Tree model.
- Research on how to create Graphical User Interface (GUI).

## 3. PROBLEMS ENCOUNTERED

- Due to low specification of hardware (laptop), it took a very long time when running the hyperparameter tuning code for Support Vector Machine model. This is because the sample of dataset is too large (30000 samples).

## 4. SELF EVALUATION OF THE PROGRESS

- The progress is good.

Supervisor's signature

Student's signature

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR



WEEKLY REPORT

<b>Trimester, Year: Trimester 3, Year 3</b>	<b>Study week no.: 6</b>
<b>Student Name &amp; ID: Hee Tuck Hoe, 19ACB03841</b>	
<b>Supervisor: Dr. Lim Jia Qi</b>	
<b>Project Title: Predictive Modeling in credit risk assessment</b>	

**1. WORK DONE**

- Gradient Boosting Decision Tree model is developed.
- Compare all the models and choose the best model to be implemented into the GUI.

**2. WORK TO BE DONE**

- Develop the GUI.

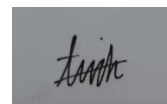
**3. PROBLEMS ENCOUNTERED**

**4. SELF EVALUATION OF THE PROGRESS**

- The progress is good.



Supervisor's signature



Student's signature

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

WEEKLY REPORT

<b>Trimester, Year: Trimester 3, Year 3</b>	<b>Study week no.: 8</b>
<b>Student Name &amp; ID: Hee Tuck Hoe, 19ACB03841</b>	
<b>Supervisor: Dr. Lim Jia Qi</b>	
<b>Project Title: Predictive Modeling in credit risk assessment</b>	

**1. WORK DONE**

-GUI is created and tested.

**2. WORK TO BE DONE**

-Work on the report.

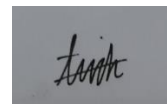
**3. PROBLEMS ENCOUNTERED**

**4. SELF EVALUATION OF THE PROGRESS**

-The progress is good.



\_\_\_\_\_  
Supervisor's signature



\_\_\_\_\_  
Student's signature

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

WEEKLY REPORT

<b>Trimester, Year: Trimester 3, Year 3</b>	<b>Study week no.: 10</b>
<b>Student Name &amp; ID: Hee Tuck Hoe, 19ACB03841</b>	
<b>Supervisor: Dr. Lim Jia Qi</b>	
<b>Project Title: Predictive Modeling in credit risk assessment</b>	

**1. WORK DONE**

-Report still in progress.

**2. WORK TO BE DONE**

-Complete the report.

-Prepare presentation slide for the presentation of Final Year Project 2 on week 13 Friday.

**3. PROBLEMS ENCOUNTERED**

**4. SELF EVALUATION OF THE PROGRESS**

-The progress is good.

Supervisor's signature

Student's signature

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

WEEKLY REPORT

<b>Trimester, Year: Trimester 3, Year 3</b>	<b>Study week no.: 13</b>
<b>Student Name &amp; ID: Hee Tuck Hoe, 19ACB03841</b>	
<b>Supervisor: Dr. Lim Nia Qi</b>	
<b>Project Title: Predictive Modeling in credit risk assessment</b>	

**1. WORK DONE**

- Report completed.
- Presented the project to moderator and supervisor.

**2. WORK TO BE DONE**

- Make final refinement on report based on moderator's suggestion and submit the report before deadline.

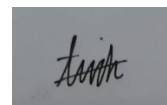
**3. PROBLEMS ENCOUNTERED**

**4. SELF EVALUATION OF THE PROGRESS**

- The progress is good.



Supervisor's signature




Student's signature

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

# POSTER



**UTAR**  
UNIVERSITI TUNKU ABDUL RAHMAN

## FACULTY OF INFORMATION COMMUNICATION AND TEHNOLOGY

### PRERICTIVE MODELING IN CREDIT RISK ASSESMENT

---

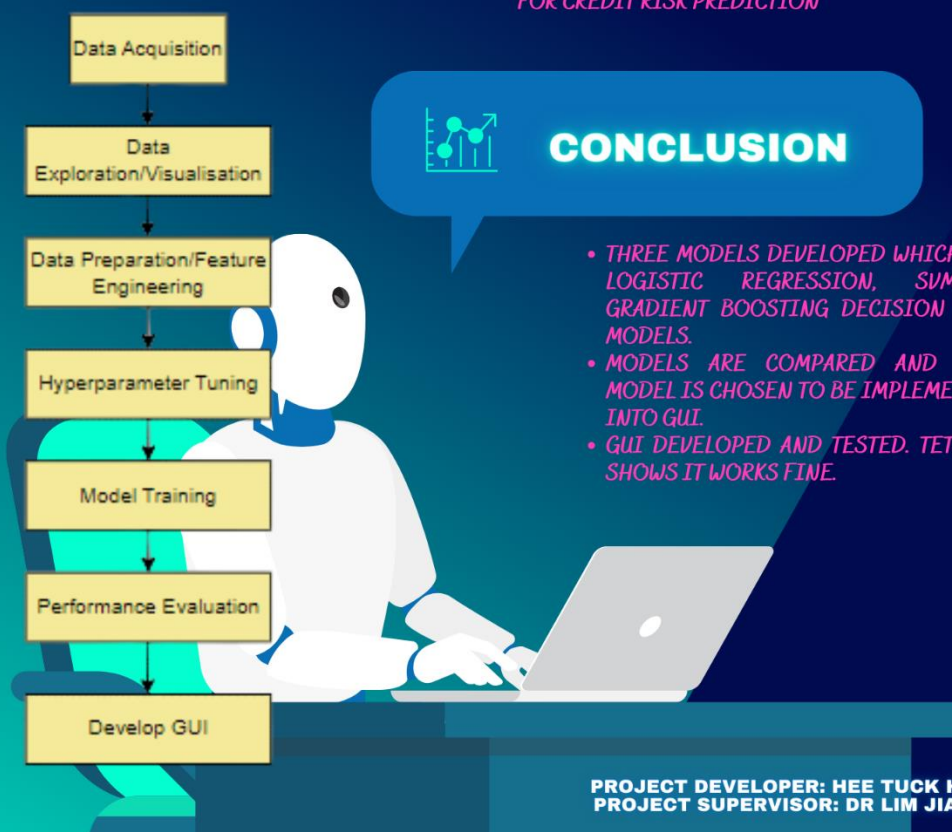
### INTRODUCTION

CREDIT RISK IS THE POSSIBILITY OF SUFFERING A LOSS AS A RESULT OF A BORROWER'S FAILURE TO MAKE LOAN PAYMENTS OR FULFIL CONTRACTUAL OBLIGATIONS. CREDIT RISK MODEL IS DEVELOPED TO IDENTIFY THE CREDIT RISK. IT HELPS THE LOAN FIRM TO ACHIEVE THE VALIDATION FASTER COMPARED TO THE TRADITIONAL METHOD.

### OBJECTIVE

- TO DEVELOP A ROBUST CREDIT RISK MODEL TO IDENTIFY THE CREDIT RISK OF CUSTOMERS
- TO DEVELOP GRAPHICAL USER INTERFACE (GUI) FOR CREDIT RISK PREDICTION

### RESEARCH WORKFLOW



```

graph TD
    A[Data Acquisition] --> B[Data Exploration/Visualisation]
    B --> C[Data Preparation/Feature Engineering]
    C --> D[Hyperparameter Tuning]
    D --> E[Model Training]
    E --> F[Performance Evaluation]
    F --> G[Develop GUI]
  
```

### CONCLUSION

- THREE MODELS DEVELOPED WHICH ARE LOGISTIC REGRESSION, SVM & GRADIENT BOOSTING DECISION TREE MODELS.
- MODELS ARE COMPARED AND BEST MODEL IS CHOSEN TO BE IMPLEMENTED INTO GUI.
- GUI DEVELOPED AND TESTED. TETSING SHOWS IT WORKS FINE.

**PROJECT DEVELOPER: HEE TUCK HOE**  
**PROJECT SUPERVISOR: DR LIM JIA QI**

# PLAGIARISM CHECK RESULT

<p>Turnitin Originality Report</p> <p>Processed on: 28-Apr-2023 10:15 +08  ID: 2077382321  Word Count: 9778  Submitted: 2</p> <p>fyp hth By Tuck Hoe Hee</p>		<p>Similarity Index</p> <p><b>20%</b></p>	<p>Similarity by Source</p> <p>Internet Sources: 11%  Publications: 6%  Student Papers: 15%</p>
--	--	---	---

1% match (student papers from 08-Jan-2023) <a href="#">Submitted to University of Bolton on 2023-01-08</a>
1% match (Internet from 22-Nov-2022) <a href="http://www.diva-portal.se/smash/get/diva2:1360926/FULLTEXT01.pdf">http://www.diva-portal.se/smash/get/diva2:1360926/FULLTEXT01.pdf</a>
1% match (student papers from 10-Dec-2022) <a href="#">Submitted to Old Dominion University on 2022-12-10</a>
1% match (student papers from 15-Apr-2023) <a href="#">Submitted to Middlesex University on 2023-04-15</a>
1% match (student papers from 21-Aug-2022) <a href="#">Submitted to University of Hull on 2022-08-21</a>
1% match (Internet from 10-Aug-2020) <a href="https://scholarworks.wm.edu/cgi/viewcontent.cgi?amp=8&amp;article=6802&amp;context=etd">https://scholarworks.wm.edu/cgi/viewcontent.cgi?amp=8&amp;article=6802&amp;context=etd</a>
1% match (Internet from 23-Apr-2023) <a href="https://www.javatpoint.com/best-languages-for-gui">https://www.javatpoint.com/best-languages-for-gui</a>
1% match (student papers from 20-Mar-2023) <a href="#">Submitted to National Economics University on 2023-03-20</a>
< 1% match (student papers from 31-Mar-2023) <a href="#">Submitted to University of Bolton on 2023-03-31</a>
< 1% match (student papers from 29-Nov-2022) <a href="#">Submitted to University of Westminster on 2022-11-29</a>
< 1% match (student papers from 01-Oct-2022) <a href="#">Submitted to University of Westminster on 2022-10-01</a>
< 1% match (student papers from 15-Sep-2022) <a href="#">Submitted to University of Westminster on 2022-09-15</a>
< 1% match (student papers from 24-Jul-2022) <a href="#">Submitted to University of Westminster on 2022-07-24</a>
< 1% match (Internet from 30-Mar-2023) <a href="http://eprints.utar.edu.my/4764/1/fyp_IB_2022_FTY.pdf">http://eprints.utar.edu.my/4764/1/fyp_IB_2022_FTY.pdf</a>

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

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



## FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

<b>Full Name(s) of Candidate(s)</b>	Hee Tuck Hoe
<b>ID Number(s)</b>	19ACB03841
<b>Programme / Course</b>	UCCC3596 Project II
<b>Title of Final Year Project</b>	Predictive Modeling in credit risk assessment

<b>Similarity</b>	<b>Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)</b>
<b>Overall similarity index: <u>20</u> %</b>  <b>Similarity by source</b>  Internet Sources: <u>11</u> % Publications: <u>6</u> % Student Papers: <u>15</u> %	
<b>Number of individual sources listed of more than 3% similarity: <u>0</u></b>	
<b>Parameters of originality required, and limits approved by UTAR are as Follows:</b> (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 <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

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: Lim Jia Qi

Date: 28/4/2023

Signature of Co-Supervisor

Name: \_\_\_\_\_

Date: \_\_\_\_\_

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

# REPORT CHECKLIST



## UNIVERSITI TUNKU ABDUL RAHMAN

### FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

#### CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB03841
Student Name	Hee Tuck Hoe
Supervisor Name	Dr. Lim Jia Qi

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.
	Front Plastic Cover (for hardcopy)
✓	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)
	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: 28/4/2023

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR



## REPORT CHECKLIST

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR