PREDICTIVE MODELLING FOR STUDENT GRADES IN FYP

NG KERWIN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2021

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :

Name : Ng Kerwin

ID No. : 1600492

Date : 19/3/2021

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"PREDICTIVE MODELLING FOR STUDENT GRADES IN FYP"** was prepared by **NG KERWIN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature :

Supervisor : Hoo Meei Hao

Date : 13 April 2021

Signature : kckhor

Co-Supervisor : Khor Kok Chin

Date : 13/4/2021

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project.

Firstly, I would like to express my gratitude to my research supervisor, Dr Hoo Meei Hao for her invaluable advice, guidance, enormous patience and encouragement throughout the development of the research. Thank you for the prompt reply to emails and questions during the writing of the report.

Furthermore, I would also like to thank my co-supervisor, Dr Khor Kok Chin for his technical guidance, patience, and encouragement throughout the research development. Thank you for helping me throughout the building of the predictive model and writing of the report.

Special Thanks to Tan Wei Yan, who encouraged me not to give up when I face problems during the writing of the report. It is also him that encouraged me not to have negative thoughts towards the project's outcome and to take the writing of the project one step at a time.

Lastly, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement throughout the entire project. Without the positive support and encouragement, I will not be able to complete the project as scheduled.

ABSTRACT

Predicting students’ grade in Final Year Project is difficult because the factors may not be purely based on a student’s academic performance. The project focus on using the academic performance of students and their logbook to predict the Final Grades of students in the Final Year Project. This project aims to predict the grade of students in the Final Year Project to decrease the student’s failure, attrition and withdrawal rate. The project proposed using classification which is part of the data mining process to predict the students’ Final Year Project Grades. The proposed prediction model are K-Nearest Neighbours, CART, C4.5, Naïve Bayes, Support Vector Machine and Neural Network. The methodology adopted by the project is a modified version of CRISP-DM (Cross Industry Standard Process for Data Mining) to cater to the needs of this project. The steps include domain understanding, data collection, data understanding, data preparation, modelling and model evaluation.The project successfully created a dataset based on students’ logbook and academic data which will ease future students’ work to do predictions on FYP 2 Grades of students. Empirical studies have been performed and it is found that other than CGPA many features collected during the data collection process are found useful in predicting the Final Grades of students in the Final Year Project. It is also confirmed that the use of Support Vector Machine Model on the dataset created during the project can deliver a good outcome in predicting students FYP2 Grades.

.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

FYP	Final Year Project
UTAR	Universiti Tunku Abdul Rahman
CRISP-DM	Cross Industry Standard Process for Data Mining
KNN	K-Nearest Neighbours
SVM	Support Vector Machine
NB	Naïve Bayes
NN	Neural Network

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

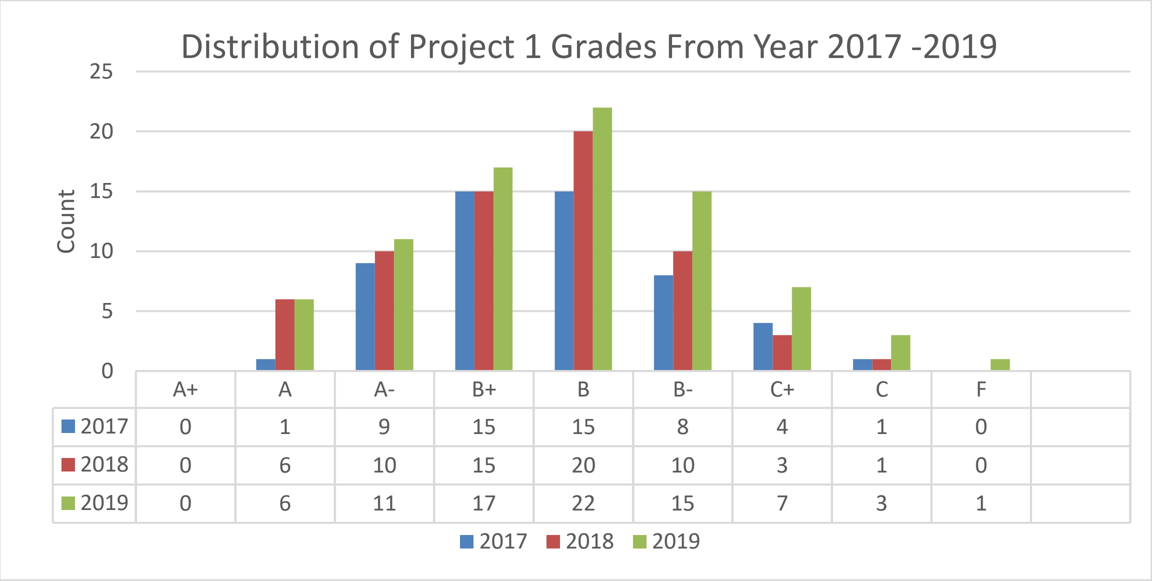
Predicting the final grades of students taking the Final Year Project (FYP) is difficult because the factors may not be purely based on the student's academic performance. The students' economic status, personal problems, and psychology may influence the predictions (Ramesh et al.,2013). Due to the limitations on gathering the data for the student, this project's prediction will focus on using logbook 1 and logbook 2 of a student during their Final Year Project and data pre-existing in the university's system to predict the grades of students taking FYP.

Data mining aims to process large amounts of data to learn the underlying patterns and link of each data towards the outcome (Kotsiantis et al.,2007). In this project, the classification task, which is part of the data mining technique to process data will be implemented in which a model will be trained on labelled data to perform predictions. The models' data is gathered manually based on individual logbooks and student's data in the system. Before predictions, data cleaning and processing were done to determine the feature to be used for predictions as not all features are relevant to the target of predictions (Langley,1994). Supervised algorithms such as K-Nearest Neighbours, CART, C4.5, Naïve Bayes, Support Vector Machine and Neural Network are to be used, and the True Positive Rate is used as an evaluation Matrix towards these models.

1.2 Problem Statement

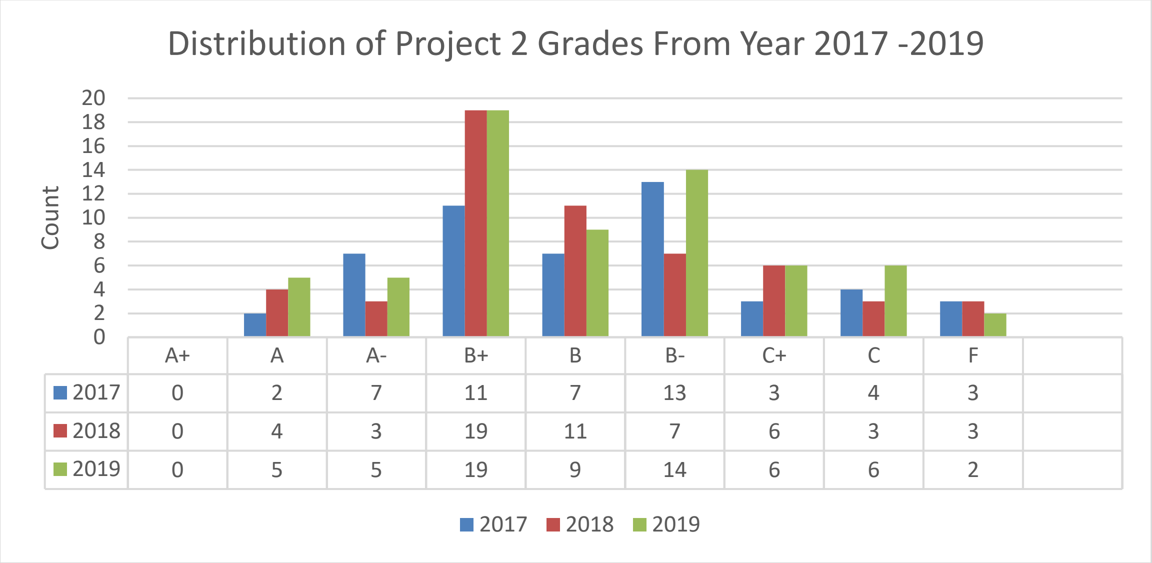
FYP in UTAR consists of 2 subjects, Project I and Project II. Students are said to pass their FYP only when they pass both the subjects. The source of the data from Figure 1.1 and Figure 1.2 is UTAR. Figure 1.1 and Figure 1.2 shows that, regardless of FYP 1 or FYP 2, the majority of the grades of students falls in the category of B+ to B-. For three years, 2019,2018 and 2017, no students can score A+ for their FYP 1 and FYP 2. Figure 1.1 and Figure 1.2 also show that the number of students who fail FYP2 is more than the number of students who Fail FYP 1.

Supervisors in UTAR aims to push most students grades from B+ to A and reduce the failure rate of FYP 2. To do that, supervisors of FYP have to acquire a way to gain knowledge on which students need assistance during their FYP to increase the majority grades of students and reduce the failure and withdrawal rate of students during FYP. Therefore, here comes the needs to predict the grades of FYP student.



	A+	A	A-	B+	B	B-	C+	C	F
🟦 2017	0	1	9	15	15	8	4	1	0
🟥 2018	0	6	10	15	20	10	3	1	0
🟩 2019	0	6	11	17	22	15	7	3	1

Figure 1.1: Bar plot of Project 1 Grade from the Year 2017-2019



	A+	A	A-	B+	B	B-	C+	C	F
🟦 2017	0	2	7	11	7	13	3	4	3
🟥 2018	0	4	3	19	11	7	6	3	3
🟩 2019	0	5	5	19	9	14	6	6	2

Figure 1.2: Bar plot of Project 2 Grade from the Year 2017-2019

There are a few problems that must be solved for predictions to be made. The first problem is the need to structure the data from individual logbooks of the student and the faculty's data. Attribute selections should also be done to ensure the attributes used in predictions can enhance the model's performance.

The discussion of each related problem statements is discussed as below:

I. Data in faculty is not structured

UTAR makes it compulsory for students in their Final Year to take FYP. During the Final Year Project, a student needs to report biweekly to their respective supervisor to update their supervisor on the project's progress. After that, the Supervisor or Co-supervisor will have to acknowledge the students' progress using either Satisfactory or Unsatisfactory. These comments by the supervisor can be used as an attribute during model training, but the problem that occurs here is that these data are not structured in a way that is easily extractable. Different students may have a different number of Satisfactory or Unsatisfactory in their logbook 1 and logbook 2. Furthermore, these comments are not counted or consolidated by the university system. Data pre-processing techniques can be applied to the students' logbook to make them structured. The technique focuses on analysing raw data to produce quality data by collecting, transforming, cleaning and summarising raw data (Zhang, S. et al.,2003).

II. The possibility that certain attributes may not be related to predictions

Attributes selection are deemed as a significant step during data mining. Adding more attributes to the model might not increase the accuracy of the model. Sometimes it might reduce accuracy. This is because the model might be overfitted or the attributes added don't correlate to the predictions to be made. In this project, many attributes will be used to build the model.

Logically thinking some attributes might not be useful in doing predictions. Attribute selection is deemed successful if the dimension is reduced and the accuracy of the predictions model remains the same or improves. The common practice nowadays is to observe the correlations of the attributes using a heatmap. A good attribute is said to have a high correlation to its target and does not highly correlate to the other features until the point that it can be predicted using another feature (Yu et al., 2003). Mutual information can also be used as an estimation to decide whether an attribute is good for predictions of the labels or not. According to Latham and Roudi (2009), high Mutual information between the feature and the label means that the feature can reduce a high amount of uncertainty in predicting the label.

III. Unable to accurately identify students that are failing

Without the aid of predictions, it is difficult to identify weak students that require assistance. These students might not know they are at risk of failing their FYP before it is too late. According to Cheng (2000), Asian students are found to be shy. This means that even when they encounter a problem, they would not speak up and seek help from their supervisor. This makes identifying weak students for supervisor harder because if a student does not ask for help, the supervisor will have to put in more effort to identify if a student requires assistance. If supervisors can intervene and provide aid earlier to the weaker students, they can improve the students’ performance during FYP (Etter et al., 2000).

1.3 Project Objectives

The problem statements discussed in 1.2 will be solved in this research project by meeting the following objectives:

1. To create a dataset with attributes that can help in predicting students' grades.
2. To identify useful attributes in a dataset for predictions.
3. To predict the grades of students taking FYP using K-Nearest Neighbours, CART, C4.5, Naïve Bayes, Support Vector Machine and Neural Network.
4. To select the best predictive model using True Positive Rate in Prediction of Failing Students.

1.4 Project Approach

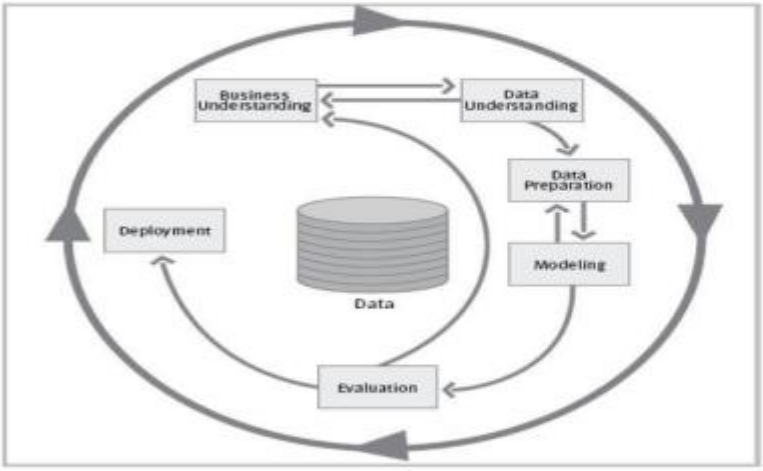


Figure 1.3: CRISP-DM Model (Yaacob et al.,2019, p2)

The proposed approach for this study is CRISP-DM (Cross Industry Standard Process for Data Mining) (Yaacob et al.,2019, p2). This approach is chosen for the data mining project because it is the most used methodology in data mining projects (Marbán, Menasalvas, Fernández-Baizán,2009). Some data mining projects on predicting students' performances also used CRISP-DM that results in high accuracy model such as the project "Supervised data mining approach for predicting student performance" resulted in all the models having an accuracy of 80% and above(Yaacob et al.,2019), "Educational Data Mining: A Hybrid Approach to Predicting Academic Performance of Students" resulted in the model having a mean error as low as 0.026 (de Almeida,2015), and "Student Performance Prediction by Using Data Mining Classification Algorithms" resulted in all the models having an accuracy of more than 70% (Kabakchieva,2012). CRISP-DM also has achieved a "factor standard" by public acceptance (Marbán, Menasalvas, Fernández-Baizán,2009).

Crisp-DM is a cyclic process that starts with understanding the project requirement. The process consists of transforming the project requirement of predicting FYP student grades into a data mining problem. The second step consists of data analysis, where data collection, data exploration, and data familiarization are done to understand the data better. General information such as the distribution of data, patterns are analysed in this step. The third steps consist of data preparation involves data transformation, data formatting and data cleaning. This is to ensure noisy data, inconsistent data, and messy data are removed to increase the model's efficiency.

Next, a data mining model is chosen, built and tuned. In this project, K-Nearest Neighbors, CART, C4.5, Naïve Bayes, Support Vector Machine and Neural Network are used as the predictive algorithms. After the model is ready, it will then go through evaluation using labelled data to check its true positive rate. The model with the highest true positive rate will be chosen for deployment.

1.5 Scope of the Project

1.5.1 Deliverables

In this project, a structured data set based on UTAR Software Engineering Students' FYP1 logbook, FYP2 logbook and academics data are created. Besides that, attributes found useful in the dataset is also recorded.

Lastly, a well-tuned classification algorithm model is developed. The chosen classification algorithm model is the model with the highest true positive among all the other various classification models used. The chosen algorithm will be trained using selected attributes that are found useful. After the model is trained, attributes found useful, best parameters for model tuning, true positive rate, AUC score, accuracy will be recorded.

1.5.2 Modules Covered

1. Prediction of FYP2 Grades of students.
2. Attributes based on FYP1 logbook, FYP2 logbook and student's academic data given by faculty.
3. Predictions Models K-Nearest Neighbours, CART, Naïve Bayes, Support Vector Machine and Neural Network are built using Python.
4. Predictions Model C4.5 is built using Weka.

1.5.3 Modules Not Covered

1. Predicting Students CGPA
2. Attributes based on students socio-economic, psychology, personal problems.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The increasing volume of educational data in the institution database has brought a need to analyse these data to discover the underlying patterns to make this abundant data useful. However, due to a large amount of data, processing and analysing the data would become a huge challenge. Educational data mining (EDM) has been found useful in handling these large amounts of data (Mueen, Zafaar, and Manzoor,2016). EDM is the process of using data mining techniques to analyse data available in the institution's database (Baker and Yacef,2009 cited in Asif,2017). EDM can be used to discover the patterns in the data. With the knowledge of these patterns, education institutions can assist students that need help. This project aims to predict the final grades of FYP students. Using the knowledge acquired from EDM, supervisors for FYP will be able to allocate resources more effectively and give assistance more frequently if needed.

There are many techniques in educational data mining. For example, classification, regression, clustering, and association rule. In this project classification tasks are to be used. Classification is a supervised learning technique used to classify data or categories. There are many classification algorithms in the arena of supervised learning such as K-nearest neighbour, Neural Network, Naïve Bayes, Decision Tree, Random Forest, and Logistic Regression. These classification algorithms must be trained with pre-existing data before they can be used for predictions and their performance is evaluated with a different evaluation matrix. Therefore,areas to be covered in the review comprise of classification algorithm,attributes related to students' performance and project methodology.

This literature review aims to:

1. Understand the classification algorithm chosen for the project.
2. Identify which classification technique is suitable for the project.
3. Identify which attributes are essential to predict students' performance.
4. Identify the methodology that should be applied in the project.

2.2 Related Works in Predicting Student Performance

The classification algorithm chosen for this project is K-Nearest Neighbours, CART, C4.5, Naïve Bayes, Support Vector Machine and Neural Network. Nine papers that uses these classification algorithms to predict student performance have been reviewed. Their respective attributes and the accuracy of the model is recorded in Table 2.1.

Table 2.1: Models, Attributes and Accuracy of Reviewed Papers

Models	Attributes	Accuracy	Author
KNN	Students' Demographic Data, Final CGPA, Courses Enrolled Marks	84.80%	Yaacoob et al. (2019)
	Admission Mark from High School, Final Marks for 1st and 2nd Year Courses	74.04%	Asif et al. (2017)
	Students' Demographic Data, Secondary School Performance Data	74.42%	Lenin and Chandrasekaran (2019)
ID3	Students' Demographic Data, Final CGPA and Courses Enrolled Marks	82.15%	Yaacoob et al. (2019)
	Admission Mark from High School, Final Marks for 1st and 2nd Year Courses	69.23%	Asif et al. (2017)
	Students' Demographic Data, Students' Enrolment Data, Students' Maths Level	95.9%	Saheed et al. (2018)
C4.5	GPA of all Subjects, Test Average Marks, Assignment Submission Status, Participation Rate in Discussion, Attendance, Lab Test Average Marks, Final Exam Marks	80.5%	Mueen, Zafaar and Manzoor (2016)
	Students' Demographic Data, Students' Enrolment Data, Students' Maths Level	98.3%	Saheed et al. (2018)
CART	Students' Demographic Data, Final CGPA, Courses Enrolled Marks	80.99%	Yaacoob et al. (2019)
	Admission Mark from High School, Final Marks for 1st and 2nd Year Courses	68.27%	Asif et al. (2017)
	Students' Demographic Data, Students' Enrolment Data, Students' Maths Level	98.3%	Saheed et al. (2018)

Naive Bayes	Students' Demographic Data, Final CGPA, Courses Enrolled Marks	89.26%	Yaacoob et al. (2019)
	GPA of all Subjects, Test Average Marks, Assignment Submission Status, Participation Rate in Discussion, Attendance, Lab Test Average Marks, Final Exam Marks	85.7%	Mueen, Zafaar, and Manzoor (2016)
	Admission Mark from High School, Final Marks for 1st and 2nd Year Courses	83.65%	Asif et al. (2017)
	Students' Demographic Data, Secondary School Performance Data	90.91%	Lenin and Chandrasekaran (2019)
Neural Network	GPA of all Subjects, Test Average Marks, Assignment Submission Status, Participation Rate in Discussion, Attendance, Lab Test Average Marks, Final Exam Marks	81.4%	Mueen, Zafaar, and Manzoor (2016)
	Admission Mark from High School, Final Marks for 1st and 2nd Year Courses	62.5%	Asif et al. (2017)
	Grades of all Courses	93.04%	Bahadir (2016)
	Students' Usage on Wiki Data, Number of Files Viewed by Students, Number of Quiz Taken by Student	98.3%	Zacharis (2016)
SVM	Math, Reading and Writing Score, Gender, Race, Parental's Education Level, Lunch Type, Test Preparation Status	90.1%	Naicker, Adeliyi, and Wing (2020)
	Students' Demographic Data, Students' Academic Data, Students' Family Data, Students' Lifestyle Data, Students' Educational Support Data,	89.74%	Athani,Kodli, Banavasi, and Hiremath (2017)

I. Attributes Used in Related Works

As shown in Table 2.1, the category of attributes used can be divided into attributes based on student academic performance and student demographic data. The most frequently used attributes are attributes related to a student's academic performance. All the papers reviewed have attributes related to student performance.

From the nine reviewed papers, five papers have been found to use the Grades of students as their main attributes to predict student performances. For instance, Yaacoob et al. (2019) used the Grades of the Courses enrolled by students. Mueen, Zafaar, and Manzoor (2016) used the marks from quizzes, lab test and assignments. Bahadir (2016) used the marks of 11 courses enrolled by the student. Asif et al. (2017) used admission marks and final marks of 1st and 2nd-year courses of students. Athani, Kodli, Banavasi, and Hiremath (2017) used first, second and final period grade.

Yaacoob et al. (2019), and Mueen, Zafaar, and Manzoor (2016) also uses CGPA or GPA to predict student performances. Furthermore, Naicker, Adeliyi and Wing (2020) also use math, writing and reading scores which is also an indicator of students' academic performance.

Five papers had been found to use student demographic data for prediction, for instance, Yaacoob et al. (2019), Lenin and Chandrasekaran (2019), Saheed et al. (2018), Athani, Kodli, Banavasi, and Hiremath(2017) and Naicker, Adeliyi and Wing (2020). Lenin and Chandrasekaran (2019) and Mueen, Zaafar, and Manzoor (2016) have found that student demographic data is not important in predicting the students' performances.

Lenin and Chandrasekaran (2019) have ranked the importance of the attributes using Boruta Library in R and Gini Index in the Random Forest Algorithm and concluded that Hperform and MBTI are the most influencing attributes. Hperform is the students' performance at the higher secondary as provided by the Board of Examination at 12th std and MBTI being the Myers-Briggs Type Indicator. Mueen, Zaafar, and Manzoor (2016) have also done feature selection using the feature selection algorithm in WEKA and found that the best seven attributes are all attributes related to student performance in academics. These seven features are GPA, Average Test Marks, Assignment submission status, Participation Rate in the discussion, Attendance, Average Lab Test Marks and Final Examination Marks. Lastly, from the result of Naicker, Adeliyi and Wing (2020), they have also found that parental level of education does not help in the prediction of students performance, whereas other attributes collected such as student performance data, race, gender is important in the prediction of student performances.

II. Performance and Algorithm Used in Related Works

As shown in Table 2.1 the reviewed papers are published in 2016-2020. During these years, the famous algorithms researchers use to do prediction on student performances are K-Nearest Neighbours, Decision Tree (ID3, CART, C4.5), Naïve Bayes, Neural Network and Support Vector Machine.

Yaacoob et al. (2019), Asif et al. (2017), and Lenin and Chandrasekaran (2019) used K-Nearest Neighbours to do predictions in their project. Yaacoob et al. (2019) measured the accuracy of the model by changing the value of K from 1 to 10. The best accuracy acquired is 84.80% when the K value is 9. Asif et al. (2017) used the K value equals 1 and the accuracy acquired is 74.04%. Lastly, Lenin and Chandrasekaran (2019) have acquired an accuracy of 74.42% when the k value is 9. Shahiri, Hussain, and Rashid (2015) also found three papers that use KNN to predict student performance and KNN gave the best accuracy in all these papers.

Next, three papers are found to use the ID3 algorithm to predict student performance which is, Yaacob et al. (2019), Asif et al. (2017), and Saheed et al. (2018). Yaacob et al. (2019) applied pre-pruning with minimal gain 0.01 and minimal leaf size of 3 and produced a decision tree with 19 nodes and 16 leaves and the accuracy measured is 82.15%. Saheed et al. (2018) measured the accuracy of the ID3 algorithm at 95.9%, ID3 also ranked 2nd in terms of speed using only 0.05 second to perform its classification task in the research. Asif et al. (2017) acquired a satisfactory result of 69.23% using a minimal leaf size of 6.

Furthermore, the C4.5 algorithm is used by Mueen, Zafaar, and Manzoor (2016) and Saheed et al. (2018). Mueen, Zafaar, and Manzoor (2016) acquired an accuracy of 80.5% after attributes selections. Saheed et al. (2018) conclude the accuracy of the C4.5 algorithm at 98.3%. It is also the highest accuracy model in the paper. The speed of C4.5 ranked 1st using only 0.03 second to perform the classification task.

Besides that, the CART algorithm is used by Yaacoob et al. (2019), Asif et al. (2017), and Saheed et al. (2018). Yaacob et al. (2019) measured the accuracy at 80.99% by applying pre-pruning with pre-pruning with minimal gain 0.01 and minimal leaf size of 3 it produced a decision tree with 19 nodes and 16 leaves. Saheed et al. (2018) found the accuracy of the CART algorithm at 98.3%. Even though CART's accuracy is on par with C4.5 at 98.3%, it uses 0.58 second to perform a classification task that is 19 times more than C4.5.

After that, four papers are found to use Naive Bayes to predict student performance, for instance, Yaacob et al. (2019), Mueen, Zafaar, and Manzoor (2016), Asif et al.(2017) and Lenin and Chandrasekaran (2019). Naive Bayes is found to be the algorithm that produces the best accuracy in three out of four papers (Yaacob et al.,2019; Mueen, Zafaar, and Manzoor (2017);

Asif et al.,2017). Yaacob et al. (2019) acquired an accuracy of 89.26%, Mueen, Zafaar, and Manzoor (2016) acquired an accuracy of 85.70% and Asif et al. (2017) acquired an accuracy of 83.65%. Even though Naive Bayes in Lenin and Chandrasekaran (2019) is not the highest accuracy model, its accuracy is still higher than three of the papers with an accuracy of 90.91%.

Moreover, four papers are found to use Neural Networks for predicting student performance, for instance, Mueen, Zafaar, and Manzoor (2016), Asif et al. (2017), Bahadir (2016) and Zacharis (2019). Mueen, Zafaar, and Manzoor (2016) acquired an accuracy of 81.4%, Asif et al. (2017) acquired an accuracy of 62.5%, Bahadir (2016) acquired an accuracy of 93.04% and Zacharis (2016) acquired an accuracy of 98.3%. Shahiri, Hussain, and Rashid (2015) also found eight papers that used a neural network to predict student performance that gave satisfactory accuracy.

Lastly, two papers are found to use Support Vector Machine to predict students' performance, for instance, Naicker, Adeliyi and Wing (2020) and Athani, Kodli, Banavasi, and Hiremath(2017). Naicker, Adeliyi and Wing (2020) have acquired an accuracy of 90.1% in predicting student performance using Linear Support Vector Machine. It is the best classifier in the paper as compared to Naïve Bayes, Decision Tree and Logistic Regression. Athani, Kodli, Banavasi, and Hiremath(2017) have also acquired an accuracy of 89.74% using Support Vector Machine to predict students' grades. They concluded that SVM is a good classifier for the prediction of students' academic performance.

2.3 Review of Predictive Model Chosen

Many algorithms can be used to create a prediction model. While all of them have the same task, which is to classify the sample data, they are based on different mathematical formulas. The chosen algorithm for this project is K-Nearest Neighbours, Decision Tree, Naïve Bayes and Neural Network. This section will briefly introduces each of the chosen algorithm and the mathematical formulae used.

2.3.1 Decision Tree

A decision tree is a flowchart like structure consisting of multiple nodes and branches. Each node in the decision tree will represent a test condition for the attributes and the outcome of the test is represented by a branch. The class label of the predictions is represented by the terminal node. The most common decision tree algorithm is ID3, C4.5, and CART (Mohankumar, Amuthakkani, and Jeyamala,2016). Figure 2.1 shows an example of a Decision Tree.

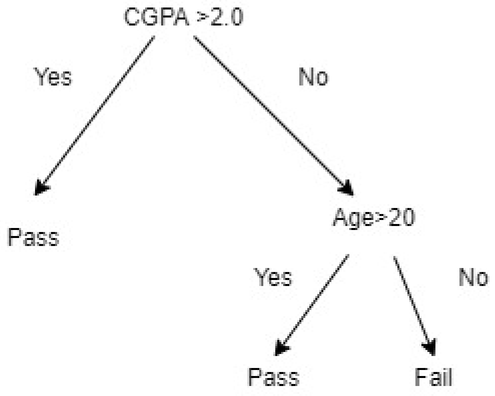


Figure 2.1: Example of Decision Tree

The tree above is used to predict the outcome of the examination of a student. The test conditions are “CGPA>2.0” and “Age>20” and the branch will show the outcome of these conditions.

I. ID3 Algorithm

ID3(Iterative Dichotomiser 3) is an algorithm created by Ross Quinlan. It uses a top-down greedy approach to build the decision tree. It means that the tree is built from top to bottom, using the greedy approach at each iteration to determine the best attributes at that present moment to create the node. The way to determine the best attributes is to calculate its information gain (Sharma and Kumar,2016). Information gains are used to measure how well a given attribute separates the data samples according to their classification. The attributes with higher information gain can remove more entropy, therefore it is chosen as the best attributes. The process is repeated until the entropy of the nodes is equal to null (Mohankumar, Amuthakkani, and Jeyamala,2016). When the entropy is equal to null the node cannot be expanded anymore because the samples in that node belong to the same class (Saheed et al.,2018).

The formula to calculate information gain and entropy is as below:

$$Gain(S|A) = Entropy(S) - (\sum \frac{|S_i|}{|S|} Entropy(S_i)) \qquad (2.1)$$

Where

$Gain(S|A)$

$= Difference\ in\ entropy\ from\ before\ to\ after\ the\ set\ S\ is\ spilt\ on\ attribute\ A$

$\sum \frac{|S_i|}{|S|} = the\ propotion\ of\ the\ number\ of\ elements\ in\ S_i\ to\ the\ number\ of\ elements\ in\ set\ S$

$Entropy(S) = Entropy\ of\ set\ S$

$Entropy(S_i) = Entropy\ of\ set\ S_i$

$$Entropy(S) = -\sum p(x)\, log_2 p(x) \qquad (2.2)$$

Where

$S = the\ current\ dataset\ which\ entropy\ is\ calculated$

$x = set\ of\ classes\ in\ S$

$p(x) = the\ propotion\ of\ the\ number\ of\ element\ in\ class\ x$

The advantages of the ID3 algorithm is it can create a short tree in a short period, it can also create an understandable prediction rule from training data. When building a decision tree, the whole data set is used. It will only test enough attributes until all the data is classified. Lastly, when searching for appropriate leaf nodes, the test data is pruned, leading to fewer tests.

However, the disadvantages of ID3 is it does not work well with numeric attributes and missing values (Mohankumar, Amuthakkani, and Jeyamala,2016).

II. C4.5 Algorithm

C4.5 algorithm is created by Ross Quinlan. The tree uses a depth-first search strategy. Different from the ID3 algorithm, C4.5 Uses gain ratio as splitting criteria into its attributes (Mueen, Zafaar and Manzoor,2016). C4.5 considers all possible tests that can be used to split the data and chooses the one with the highest gain ratio.

C4.5 can work with a categorical and numerical value. When a numerical value is used, a threshold will be declared, and the numerical value is divided into values above and below the threshold. It can also work with missing attributes value because those values will not be used to do gain ratio calculations. C4.5 also prunes trees after it has been created by removing unused branches and replacing them with leaf nodes. C4.5 is said to be able to overcome the disadvantages of the ID3 Algorithm because missing attribute values are not used during gain calculations (Sharma and Kumar,2016). The disadvantage of C4.5 is it creates empty branches. Some researchers have found that many nodes with zero values or close to zero values are created after their project. These nodes not only did not help in the classification task or rules generation, but it makes the tree more complex and bigger (Mohankumar, Amuthakkani, and Jeyamala,2016).

The formula for the Gain ratio is as below:

$$GainRatio = \frac{Gain(S,A)}{Entropy(S,A)} \tag{2.3}$$

Where

$Gain(S,A) = Information\ gain\ of\ attributes\ A\ on\ set\ S$

$Entropy(S,A) = Entropy\ attributes\ A\ on\ set\ S$

III. CART algorithm

CART (Classification and Regression Tree) algorithm is created by Leo Bierman (Breiman et al.,1984 cited in Mohankumar, Amuthakkani, and Jeyamala,2016). Unlike ID3 and C4.5, CART uses a Gini index as its splitting criterion and produces a binary split (Sharma and Kumar,2016). It can be used in both classification and regression. The classification tree produced by CART is based on binary splitting of attributes. CART supports nominal and continuous data and the speed of processing is average. It can also deal with missing values.

The formula of the Gini Index is as below:

$$Gini\ Index = 1 - \sum_i [p\left(\frac{i}{t}\right)]^2 \tag{2.4}$$

Where

$p\left(\frac{i}{t}\right) = \ The\ fraction\ of\ records\ belonging\ to\ class\ i\ at\ a\ given\ node\ t$

2.3.2 K-Nearest Neighbours

K-Nearest Neighbours is a well-known classification method. It is known as the lazy-learning algorithm because it takes less time to train the model. The classification speed is also faster than Decision Tree and Naive Bayes (Kosiantis 2007 cited in Singh and Lakshmiganthan, 2019).

It classifies objects based on its majority vote on its neighbours on the trained set. The sample of data or object will be assigned to the class or target variable that appeared most on its k nearest neighbours. For example, if k=1, the sample data will be assigned to the class of the closest neighbours of that data (Yaacoob et al.,2019). Even though the Euclidean formula is generally used to calculate the distance between two points there are still other formulas such as Manhattan (Singh and Lakshmiganthan,2019).

$$dist(x, y) = \sqrt{\sum_{i=1}^{m} (X_i - Y_i)^2} \tag{2.5}$$

where

$x = coordinates\ of\ X$

$x = coordinates\ of\ Y$

2.3.3 Naïve Bayes

Navies Bayes algorithm is a probabilistic classifier that uses the probabilistic relationship between the attributes and their classes. Attributes are used to calculate their respective probability that belongs to a class and these attributes are finally assigned to the class in which the probability is highest. It predicts the probability of a sample that belongs to a class based on the Bayesian theorem (Mueen, Zafaar, and Manzoor,2016).

The formula of the Bayesian Theorem is as below:

$$P(A|X) = \frac{P(X|A) * P(A)}{P(X)} \tag{2.6}$$

$P(A|X) = probability\ of\ attributes\ X\ being\ in\ class\ A$

$P(X|A) = probability\ of\ getting\ attributes\ X\ in\ class\ in\ class\ A$

$P(A) = probability\ of\ class\ A$

$P(X) = probability\ of\ instances\ \ X\ occuring$

2.3.4 Neural Network

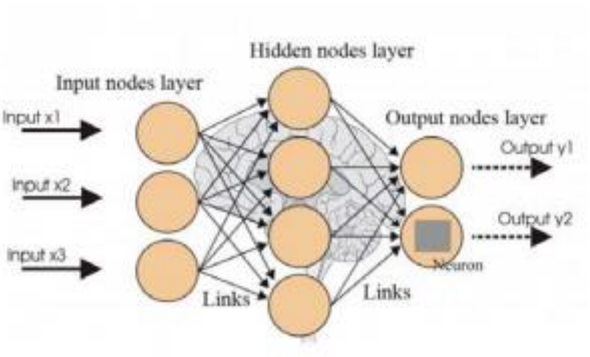


Figure 2.2: Layers of Neural Network (Mijwel,2018)

Figure 2.2 shows the layers of the Neural Network. Neural Network imitates how the animal brain works (Zhacharis2019). As shown in Figure 2.2, the neural network consists of nodes that are arranged layer by layer. Each of the nodes in a layer is connected by a channel to another node in another layer. A typical Neural Network should consist of 3 layers. An example of layers is the input layer, which accepts input, the hidden layer that processes the input, and the output layer, which shows the final predicted class. The number of hidden layers can be increased depending on the performance of the network. The input layer accepts sample data to be processed by the hidden layer and the output layer will produce the class that belongs to the sample data.

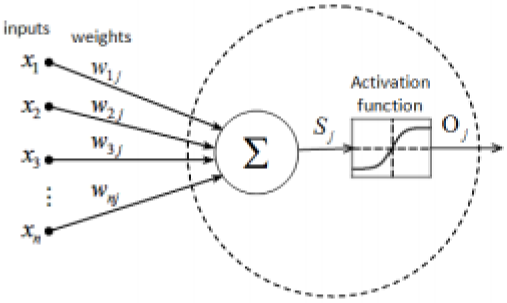


Figure 2.3: Layers of Neural Network (Mijwel,2018)

According to Figure 2.3, data received by the input layer will be multiplied by different weights. These weights can be adjusted to increase or decrease the performance of the predictive model. The resulting product will be passed to a function called activation functions, producing an output of 0 and 1. There are many activation functions available, for example, Sigmoid, TanH, SoftMax, ReLu, but the most used activation function is Sigmoid Function (Zacharis,2019). When the output reaches the threshold, the node is activated, and the product is passed to the next layer. The inactivated node will not pass the product to the next layer. The output layer will consist of only one activated neuron with the highest product and that will be our predicted class.

The sigmoid function goes by the formula:

$$o_j = \frac{1}{1 + e^{S_j}} \tag{2.7}$$

Where

$O_j = output\ of\ the\ function$

s_j is calculated by:

$$s_j = \sum_{i=1}^{n} X_i W_{ij} \tag{2.8}$$

Where

$X_i = inputs$

$W_{ij} = weight\ of\ the\ channel$

2.3.5 Support Vector Machine

The main principle of Support Vector Machine in classification is to maximise training data usage in building a classifier without overfitting the prediction model. SVM aims to create a decision boundary known as the hyperplane that maximises the margin between several classes in the training data to enable the prediction of labels with one or more feature vectors (Cervantes, Garcia-Lamont, Rodrigues-Mazahua and Lopez,2020). This hyperplane is positioned as far as possible from the closest data points in each class. The closest data points are known as support vector.

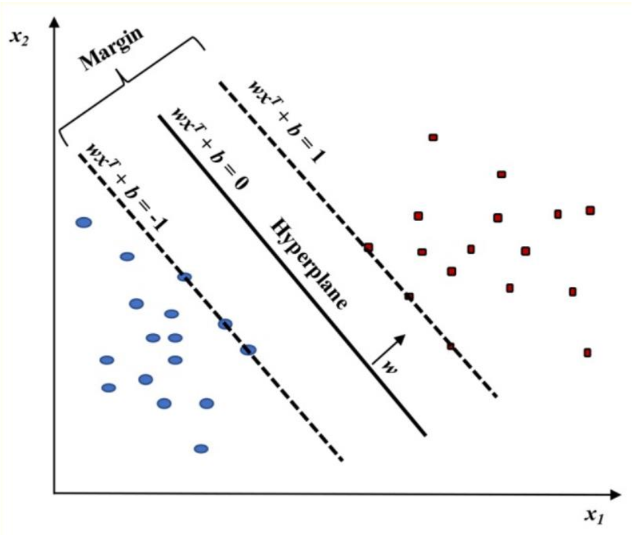


Figure 2.4: Linear SVM Model (Huang, Cai, Pacheco, Narrandes, Wang and Xu, 2018)

Figure 2.4 shows a linearly separable case. Where the X_1 and X_2 axis shown in the figure denotes the features used for predictions. As shown in Figure 2.4, the optimal Hyperplane can be denoted as $wx^T+b=0$,where w is the weight and x are indicated the bias. When training an SVM classifier, the aim is to find the best value for the weight and the bias that maximises the margin between 2 classes from their support vector. A data point is considered a support vector when the distance, $|Y_i|$ (wx^T+b) =1, where Y_i denotes the closest data point of each class, red and blue.

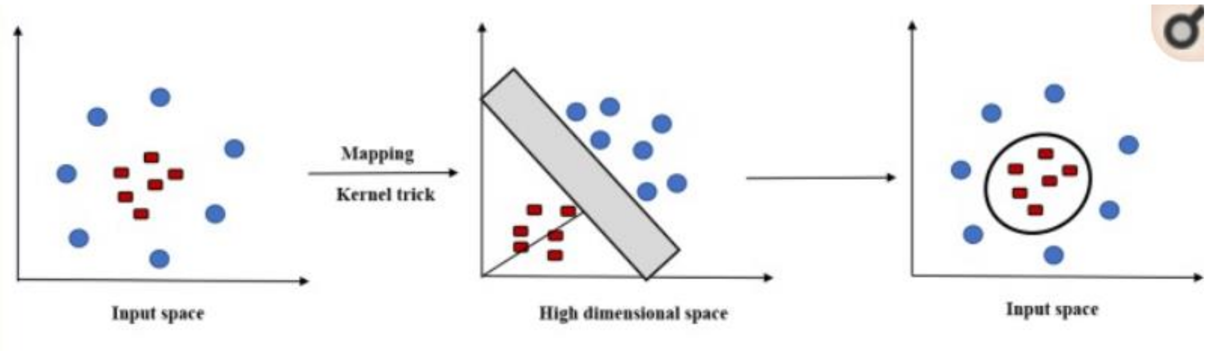


Figure 2.5: Transformation of Non-Linear to Linear using Kernel Function (Huang, Cai, Pacheco, Narrandes, Wang and Xu, 2018)

However not all classes as linearly separable. For non-linearly separable classes, a kernel function must be used to add a dimension that transforms the non-linearly separable classes to linearly separable classes in a higher-dimensional space. Figure 2.5 above shows the process of transforming a nonlinear separable class to linearly separable classes with the help of a kernel function. There are different types of Kernel function, namely Linear, Polynomial, Gaussian, RBF and Sigmoid with different computation (Cervantes, Garcia-Lamont, Rodrigues-Mazahua and Lopez,2020). There is no unanimous conclusion in which kernel is better, therefore the best way to find the best kernel for each problem is to try them all out.

2.4 Cross-Industry Standard Process for data mining (CRISP-DM)

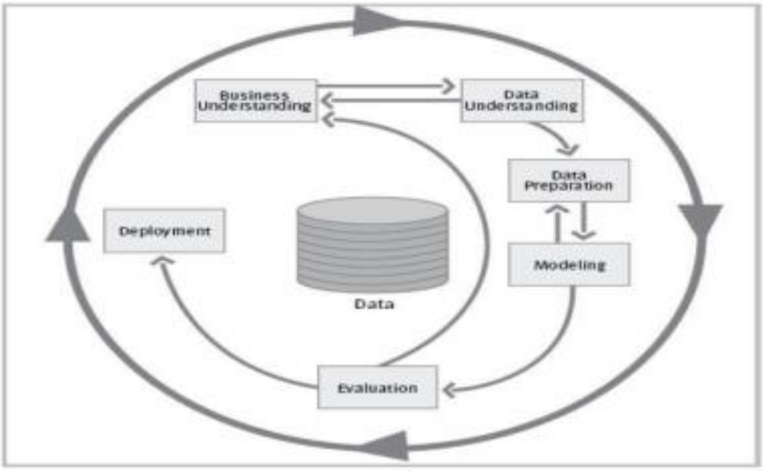


Figure 2.6: CRISP-DM Model (Yaacob et al.,2019, p2)

The methodology reviewed is called Cross-Industry Standard Process for Data Mining (CRISP-DM). There are 6 general steps of CRISP-DM consisting of business understanding followed by data understanding (Clark,2018). Then data preparation is done before modelling and evaluation. If the model performance is satisfied, the model can be deployed. The strength of this methodology is that it can be modified to the needs of the project (Clark,2018).

The first step is Business/Domain Understanding. In this step, the requirement, objectives and scope of the project will be determined. On top of that, the software and language used, attribute used, the algorithm used in the project is determined. Using all the details determined, a preliminary plan will be prepared to initiate the project. As shown in Figure 2.6, this phase will be revisited after the preliminary plan is evaluated to improve the project further or modify the project’s requirement and objectives.

The next step is data understanding. This is a crucial part of a project because an accurate model cannot be built without a proper understanding of the data used (Clark,2018). After the data is collected or retrieved, the correlation of the attributes to the label can be examined to understand how different variables respond to the target classes. Activities here aims to let researchers familiarise themselves with the data so that the first insight into the raw data can be discovered. According to Figure 2.4, if the data collected does not relate to the project requirement, this phase can be reverted to the previous step to re-examine the data needed for the project.

In data preparation, processes such as data collection, feature selection, data scaling, and data transformation are performed. For example, Almahadeen, Akkava, and Sari (2017) adopted CRISP-DM into their project. In data preparation steps, data collection and feature selection are done. Data are collected using questionnaires since the questionnaires consist of 15 attributes. The suitable attributes must be selected for building the model. Therefore, feature selection is done where the result excludes 6 attributes from the 15 attributes collected. Clark (2018) mentioned that in this phase, data is processed into a suitable form. Most of the structured data do not need to be processed. However, this is not the case for unstructured data. For example, a regex command can be used to extract the IP address from a log file. Data also must be scaled to ensure that all the attributes belong to the same scale. For instance, by calculating the means and standard deviations of a set of Celsius and a set of Fahrenheit data, the data can be standardised.

The next step is modelling. In this step, the algorithm chosen is implemented to build a model for predictions. Since some algorithms like Neural Nets have a lot of parameters to tune, the model will be tuned here to get the best performance of the model (Zacharis,2019). Some algorithms, for example, K-Nearest Neighbours, do not work well with categorical data. According to Figure 2.4, the project can loop back to the previous steps of data preparation to transform the data into numerical value if needed.

The next step will be the evaluation. The model is usually tested for accuracy in this phase. According to Clark (2018), a model may still not be meeting goals even though it has extremely high accuracy. Therefore, the person in charge of the evaluation should create more data points to look for the model’s unintended outcome. Almahadeen, Akkava, and Sari (2017), also evaluated their model using accuracy and found that the result was not satisfactory due to the bad attributes collected during questionnaires. Therefore, in future works, they suggested collecting more reliable information to enhance model quality. After all the steps, if the model performance and objectives of the project are satisfied, the model can be deployed. Else, the project can be reverted to the first phase which is business understanding to redetermine the ways to improve the quality of the project.

A few of the similar paper has also used CRISP-DM or a modified version of CRISP-DM in their project. For example, Yaacoob et al. (2019), Bahadir (2016), Lenin and Chandrasekaran, (2019), Mueen, Zaafar, and Manzoor (2016) and Saheed et al. (2018). Where their project includes the steps of domain understanding, data preparation, modelling and evaluation.

2.5 Summary

CGPA and score are indicators of how much potential a student has in academic because it has a tangible value that can be measured easily (bin Mat et al.,2013 cited in Shahiri, Hussain and Rashid 2015). Papers that use attributes based on student performance has also gotten a satisfactory result, therefore, these attributes will be prioritised during data collection. Even though two out of three papers that uses student demographic data founds that these data are not important, demographic data of student will still be collected for this project (Lenin and Chandrasekaran,2019; Mueen, Zaafar, and Manzoor,2016). This is because the underlying patterns of the demographic data in UTAR have yet to be discovered. During features selections, if the attributes are found to be less important, they will be discarded.

All the reviewed algorithms will be tested in the project using the data collected. The algorithms are K-Nearest Neighbours, Decision Tree, Naïve Bayes, Support Vector Machine and Neural Network. An empirical study will be done and each of the performance of the algorithm will be recorded to find out which algorithms work best on the dataset generated for this project

CRISP-DM methodology will be adopted in the project due to its ability to cater to the needs of the machine learning problem (Clark,2018). It is also a cyclic approach that is flexible which enables researchers to go back to previous steps if needed. By Applying CRISP-DM, the project is performed in systematic steps to negate the possibility of the project going astray.

CHAPTER 3

METHODOLOGY

3.1 Steps Adopted in the Project

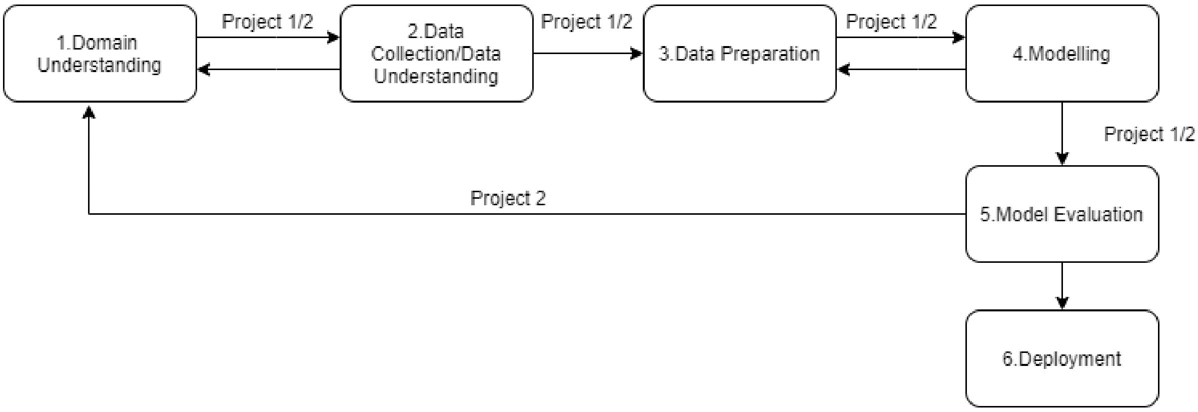


Figure 3.1: Steps of CRISP-DM implemented in the Project

Figure 3.1 is a CRISP-DM methodology modified to fit the needs of this project. The steps for this project are as below:

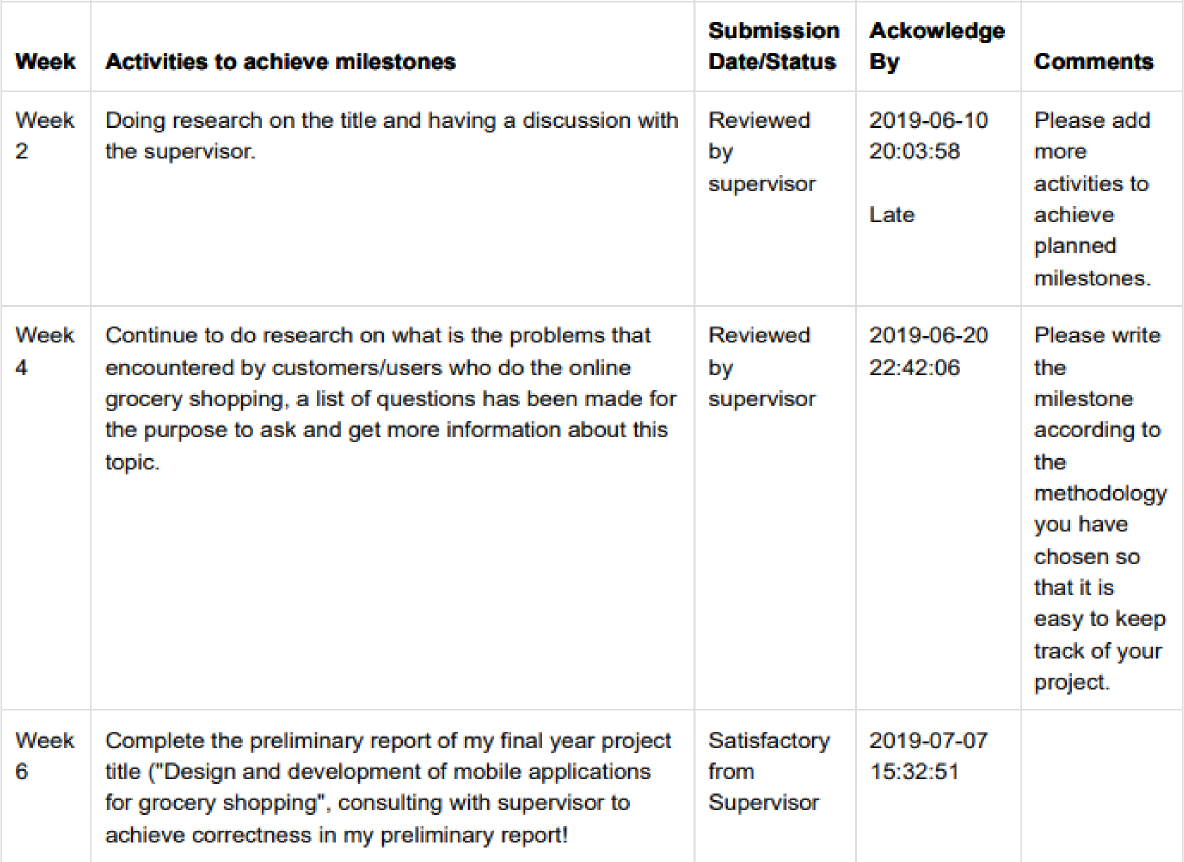
1. Domain Understanding

The steps here include understanding the title of our project, determining the project's objective and scope, and the success criteria of the project. The title of the project is "Predictive Modelling for student grade in FYP". The objective of the project is concluded to be creating a dataset with attributes that can help in predicting student's grades, to identify useful attributes in a dataset for predictions, to predict the grade of FYP student using various predictions model and lastly to select the best predictive model using true positive rate. The literature review has also been done to review previous related works to identify the model performance and attributes those related works has used. The chosen algorithm for the project is K-Nearest Neighbours, CART, C4.5, Naive Bayes, Neural Network and Support Vector Machine.

2. Data Collection / Data Understanding

I. Data Collection

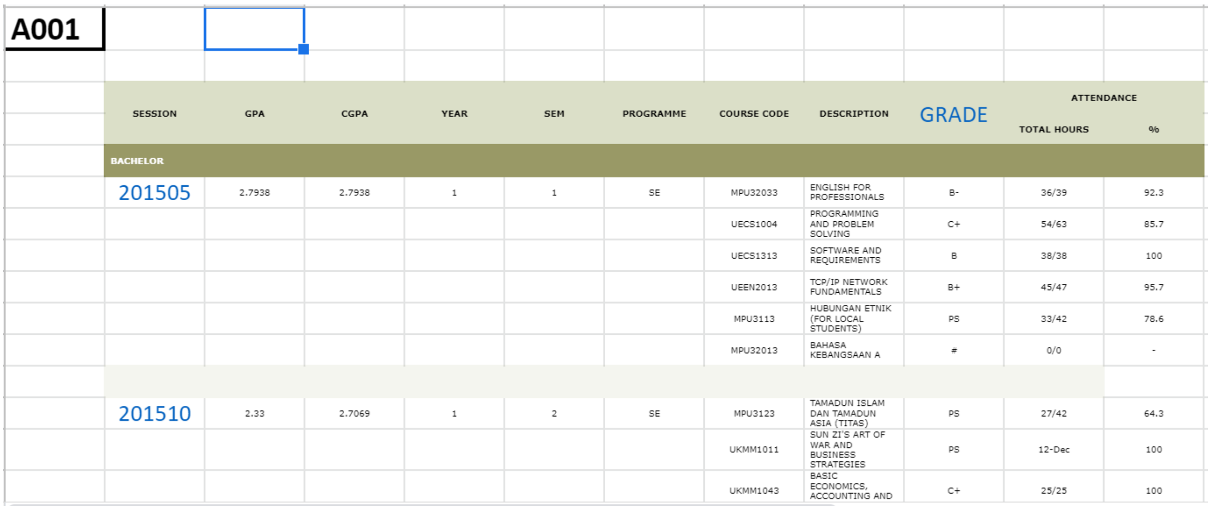
The data that had been collected for this project consists of Students' Demographic Data, Students' Academic Data and Students' Logbook Data. Students' Demographic Data is retrieved from the university database while the Students' Logbook Data are calculated and recorded manually by counting the Total Number of Logs Submission for each Logbook 1 and Logbook 2 of the student. Only Students Data from 2018 and 2019 is collected as the data for other years are lost. Figure 3.2 shows an example of Logbook 1 where the Total Number of Submission in the Logbook is 3.



Week	Activities to achieve milestones	Submission Date/Status	Ackowledge By	Comments
Week 2	Doing research on the title and having a discussion with the supervisor.	Reviewed by supervisor	2019-06-10 20:03:58 Late	Please add more activities to achieve planned milestones.
Week 4	Continue to do research on what is the problems that encountered by customers/users who do the online grocery shopping, a list of questions has been made for the purpose to ask and get more information about this topic.	Reviewed by supervisor	2019-06-20 22:42:06	Please write the milestone according to the methodology you have chosen so that it is easy to keep track of your project.
Week 6	Complete the preliminary report of my final year project title ("Design and development of mobile applications for grocery shopping", consulting with supervisor to achieve correctness in my preliminary report!	Satisfactory from Supervisor	2019-07-07 15:32:51	

Figure 3.2: Example of Logbook

The faculty gave the Students' Academic Data in the form of screenshots where the needed attributes for predictions has to be calculated and recorded manually into another excel file. The attributes that can be retrieved from the screenshot in Figure 3.3 is CGPA before FYP1, CGPA before FYP2, Industrial Training Process Before Taking Project 1, Industrial Training Process Before Taking Project 2, Number of Trimester before FYP1, Number of Trimester before FYP2, Total Number of Retakes, Total Number of Listing, Total Number of subjects failed, FYP1 Grades and FYP 2 Grades.



A001

SESSION	GPA	CGPA	YEAR	SEM	PROGRAMME	COURSE CODE	DESCRIPTION	GRADE	ATTENDANCE	
									TOTAL HOURS	%
BACHELOR										
201505	2.7938	2.7938	1	1	SE	MPU32033	ENGLISH FOR PROFESSIONALS	B-	36/39	92.3
						UECS1004	PROGRAMMING AND PROBLEM SOLVING	C+	54/63	85.7
						UECS1313	SOFTWARE AND REQUIREMENTS	B	38/38	100
						UEEN2013	TCP/IP NETWORK FUNDAMENTALS	B+	45/47	95.7
						MPU3113	HUBUNGAN ETNIK (FOR LOCAL STUDENTS)	PS	33/42	78.6
						MPU32013	BAHASA KEBANGSAAN A	#	0/0	-
201510	2.33	2.7069	1	2	SE	MPU3123	TAMADUN ISLAM DAN TAMADUN ASIA (TITAS)	PS	27/42	64.3
						UKMM1011	SUN ZI'S ART OF WAR AND BUSINESS STRATEGIES	PS	12-Dec	100
						UKMM1043	BASIC ECONOMICS, ACCOUNTING AND	C+	25/25	100

Figure 3.3: Example of Screenshot for Academic Result

II. Summary of Data Collected

A total of 263 rows and 15 attributes are collected for this project. The attributes name, type and description are shown in table 3.1.

Table 3.1: Summary of Attributes Name and Type Collected

No	Attributes	Type	Description
1	CGPA before FYP1	Numeric: from 2.0 to 4.0	CGPA before Semester that student take FYP1
2	CGPA before FYP2	Numeric: from 2.0 to 4.0	CGPA before Semester that student take FYP2
3	Industrial Training Status Before Project 1	Binary: Done (1) or Not (0)	Has the student undergo an internship before Project 1?
4	Industrial Training Status Before Project 2	Binary: Done (1) or Not (0)	Has the student undergo an internship before Project 2?
5	No of Subjects taken with Project I	Numeric: from 0 to 6	Total Subjects students take during Project 1 excluding Project 1
6	No of Subjects taken with Project II	Numeric: from 0 to 5	Total Subjects students take during Project 2 excluding Project 2
7	No of Trimesters Before FYP1	Numeric: from 5 to 12	How many semesters student study before FYP1?
8	No of Trimesters Before FYP2	Numeric: from 7 to 19	How many semesters student study before FYP2?
9	Total Number of Retakes	Numeric: from 0 to 10	How many times did student retake Major subject?
10	Total Number of Listing	Numeric: from 0 to 7	How many times did the student get into the president or dean list?
11	Total Number of Subjects Failed	Numeric: from 0 to 12	How many times does the student fail an elective or major subject?
12	Total Logs Submission in Logbook1	Numeric: from 2 to 15	How many times the student report to their supervisor during FYP1?
13	Total Logs Submission in Logbook2	Numeric: from 0 to 9	How many times the student report to their supervisor during FYP2?
14	FYP1 Grades	Categorical: A, A-, B+, B, B-, C+, C, F, W	Final Grade obtained for FYP1
15	FYP2 Grades	Categorical: A, A-, B+, B, B-, C+, C, F, W	Final Grade obtained for FYP2

III. Data Understanding

Several boxplot, scatterplot and bar graph has been plotted by using Python seaborn and matplotlib to understand the patterns in our raw data collected. The findings through plotting the graph and plots are discussed below.

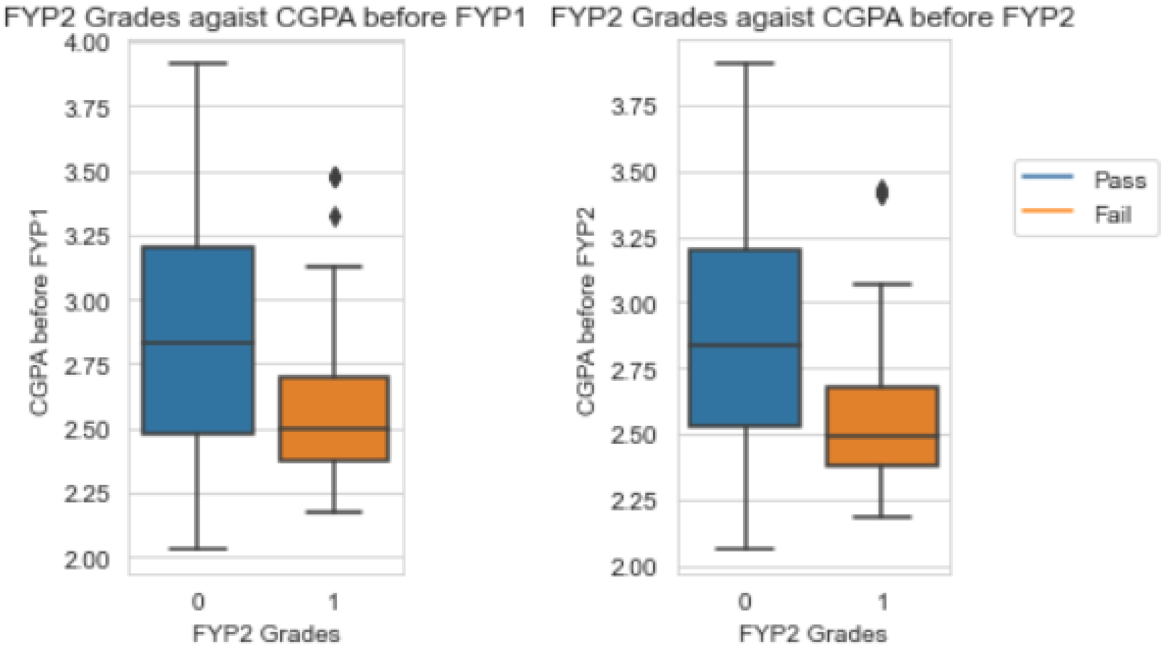


Figure 3.4: Boxplot of FYP2 Grades Against CGPA before FYP1 and FYP2

Figure 3.4 shows the boxplot of FYP2 grades against CGPA before FYP1 and FYP2,where 0 denotes passing students and 1 denotes failing students. From both of the boxplot shown above it is observed that students who failed FYP2 have a lower average CGPA before FYP1 and FYP2 compared to students that pass FYP2. The average CGPA before FYP1 and FYP2 of students that fail FYP2 is around 2.5. This means that the supervisor and co-supervisor of the Final Year Project have to keep an eye on students having a CGPA around the range of 2.5 when they take FYP1 and FYP2 because this is the group of students that is most likely to fail FYP2.

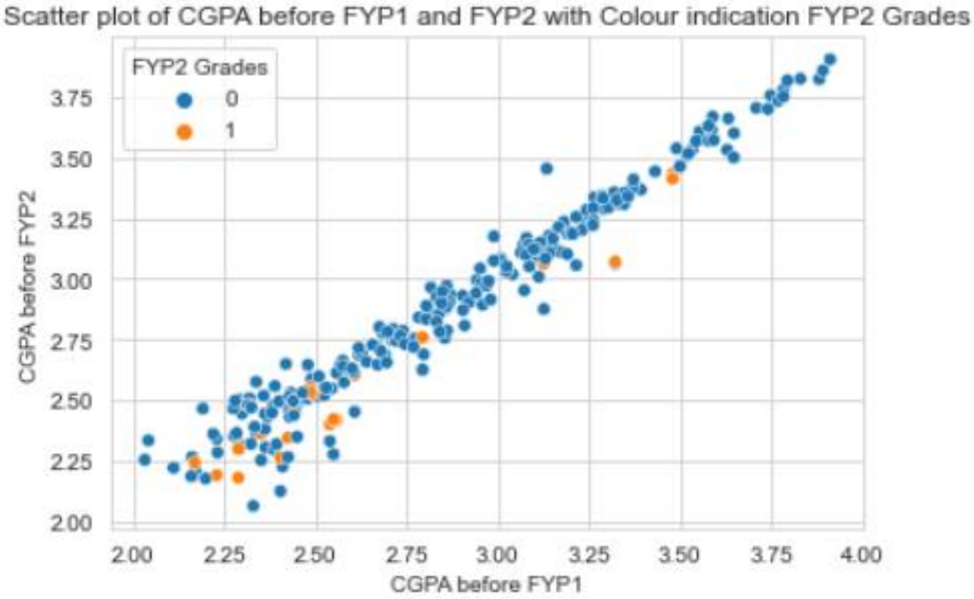


Figure 3.5: Scatterplot of CGPA before FYP1 and FYP2 with FYP2 Grades as Hue

Figure 3.5 shows the scatterplot of CGPA before FYP 1 and FYP2 with FYP2 Grades as Hue of the students in the collected dataset. 1 Denote failing students which is coloured in orange and 0 denote passing students which is coloured in blue. It is observed that majority of students who failed FYP2 has a lower CGPA before FYP1 and FYP2. There are a minority of students that have high CGPA before FYP1 as FYP2 that end up failing their FYP2. This means that other factors other than CGPA before FYP1 and FYP2 affects students to fail their FYP2.

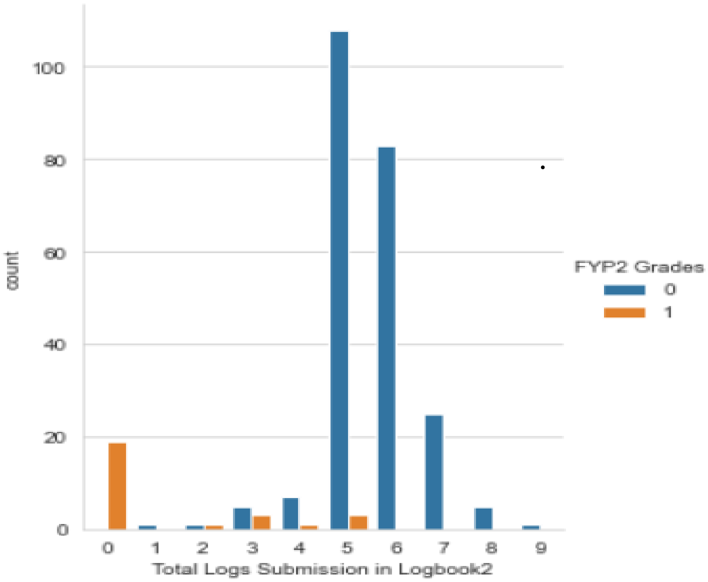


Figure 3.6: Bar Graph of Total Logs Submission in Logbook 2 with FYP2 Grades as hue

Figure 3.6 shows the bar graph of total logs submission in Logbook 2 with FYP2 as hue where 0 Denote pass and 1 denote failing students. The majority of people who pass FYP2 reports frequently to their supervisor and update their logbook frequently. The majority of failing student did not update and report to their supervisor. Some of the students who had failed report less than 5 times to their supervisor. A hypothesis can be made that the supervisor should note the frequency of students reporting to them because if a student does not report to them, they might be at risk of failing their FYP2. Students might be afraid to report their progress to the supervisor because they have not done or do not know how to proceed on their FYP2. However, some students manage to pass FYP2 with a low submission of Logs, this indicates that this feature might not be the best attributes in predicting final grades of students. Some students might have bad time management and ended up not submitting their logs but they are able to complete the FYP2 in time.

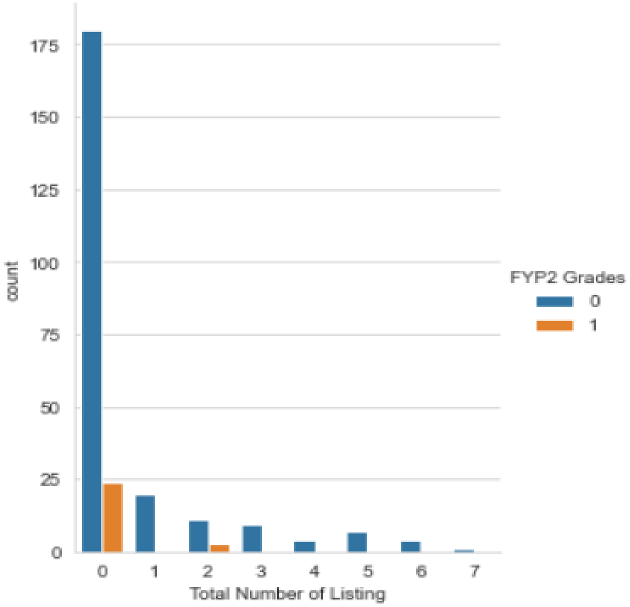


Figure 3.7: Bar Graph of Number of Listing with FYP2 grades as hue

Figure 3.7 shows the Bar Graph of Number of Listing with FYP2 grades as hue. The number of listing indicates the number of time students acquires Dean or President List in each of their long semester where their credit hour is more than 12. It is observed that majority of students that fail FYP 2 did not acquire any listing in their academics. This means that the supervisor and co-supervisor of the Final Year project need to focus more on students who have not entered any List as these are students who most likely will fail FYP2. Figure 3.7 also shows that there are students who still fail their FYP2 even though they are listed twice, this indicates that’s there might be other factors that affect students to fail their FYP2.

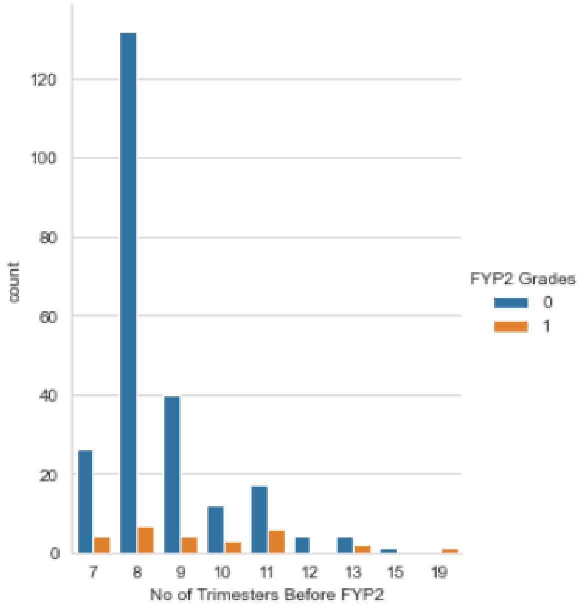


Figure 3.8: No of Trimester before FYP2 with FYP2 grades as hue

Figure 3.8 shows the No of Trimester before FYP2 with FYP2 grades as hue where 0 denotes passing students and 1 denotes failing student. It is observed that the majority of the students in the data collected took FYP 2 after studying for 8 semesters. A hypothesis can also be made that the course structure of UTAR will allow students to take FYP2 after studying for 7 to 9 semesters.

3. Data Preparation

I. Data Conversion

All the columns in the dataset are checked to make sure that none of the columns is left empty. To decrease the unique feature values of the "CGPA before FYP1" and "CGPA before FYP2", these two feature values are converted from numeric to categorical based on the CGPA cut-off point of UTAR honours classification. Where 2.0-2.99 is Normal Class, 3.0 to 3.6699 2nd class and 3.67 and above is 1st class. FYP1 and FYP2 Grades are also converted from categorical to binary to order to reduce the number of categories and increase the sample size for students at risk. The students that obtain grade A, A-, B+, B, B-, C+, C are converted to Pass, and the students that obtain grade F and W(withdrawal) are converted to Fail.

Table 3.2: Feature Conversion

No	Attribute	Conversion
1	CGPA before FYP1	3.67-4.0 =1 (1st class), 3.0 -3.6699=2 (2nd class) 2.0-2.99=3 (Normal class)
2	CGPA before FYP2	3.67-4.0 =1 (1st class), 3.0 -3.6699=2 (2nd class) 2.0-2.99=3 (Normal class)
3	FYP1 Grades	A, A-, B+, B, B-, C+, C = 0 (Pass) F, W=1 (Fail)
4	FYP2 Grades	A, A-, B+, B, B-, C+, C = 0 (Pass) F, W=1 (Fail)

The final Attributes that will be used for prediction are shown in table 3.3

Table 3.3: Attributes Name and Type after Conversion

No	Attributes	Type
1	CGPA before FYP1	Categorical: 1,2, or 3
2	CGPA before FYP2	Categorical: 1,2, or 3
3	Industrial Training Status Before Project 1	Binary: Done (1) or Not (0)
4	Industrial Training Status Before Project 2	Binary: Done (1) or Not (0)
5	No of Subjects taken with Project I	Numeric: from 0 to 6
6	No of Subjects taken with Project II	Numeric: from 0 to 5
7	No of Trimesters Before FYP1	Numeric: from 5 to 12
8	No of Trimesters Before FYP2	Numeric: from 7 to 19
9	Total Number of Retakes	Numeric: from 0 to 10
10	Total Number of Listing	Numeric: from 0 to 7
11	Total Number of Subjects Failed	Numeric: from 0 to 12
12	Total Logs Submission in Logbook1	Numeric: from 2 to 15
13	Total Logs Submission in Logbook2	Numeric: from 0 to 9
14	FYP1 Grades	Binary: 0 or 1
15	FYP2 Grades	Binary: 0 or 1

II. Dataset Preparation

According to Alazzam, Sharieh and Sabri (2020), features selection is important in machine learning as irrelevant features may decrease the accuracy of the model and increase the time needed to train the model. Therefore, the Scikit-Learn feature selection library mutual_info_classif is used to estimate the mutual information between the feature and the target (FYP2 Grades) for feature selection.

According to Latham and Roudi (2009), mutual information is a quantity that measures the relationship between two random variables. It tells us how much information can be retrieved on a random variable using one random variable. In the project, the amount of information a feature collected can tell us about the label (FYP2 Grades) is measured. A high mutual information value reduces a high amount of uncertainty in predicting the label while a low Mutual information value reduces a very low amount of uncertainty. A mutual information value of zero indicates that both the variables are independent. Table 3.4 shows the Feature and its Mutual Information.

Table 3.4: Table of Feature and respective Mutual Information Ranked

Rank	Feature	Mutual Information (2 Decimal Places)
1	Total Logs Submission in Logbook 2	0.24
2	Total Number of Retakes	0.04
3	Total Number of Subjects Failed	0.04
4	No of Trimesters Before FYP2	0.04
5	No of Subjects taken with Project II	0.03
6	Total Number of Listing	0.02
7	Total Logs Submission in Logbook1	0.02
8	No of Trimesters Before FYP1	0.01
9	CGPA before FYP2	0.01
10	CGPA before FYP1	0.01
11	No of Subjects taken with Project I	0.01
12	Industrial Training status before Project 2	0.00
13	Industrial Training status before Project 1	0.00
14	FYP1 Grades	0.00

Based on the Mutual Information obtained 14 Datasets have been created to see the model performance whenever a feature is added. The model performance on the 14 Datasets has been evaluated and recorded to see which features does not improve the performances of the model. The 14 datasets and their attributes are shown below:

Table 3.5: Datasets

Dataset	Features Ranked (Based on Table 3.4)	Number of Features
1	1	1
2	1 to 2	2
3	1 to 3	3
4	1 to 4	4
5	1 to 5	5
6	1 to 6	6
7	1 to 7	7
8	1 to 8	8
9	1 to 9	9
10	1 to 10	10
11	1 to 11	11
12	1 to 12	12
13	1 to 13	13
14	1 to 14	14

4. Model Building

I. Python

The model that has been built using python Scikit-Learn library is K-Nearest Neighbours, CART, Naïve Bayes, Support Vector Machine and Neural network. The ID3 Decision Tree is not used in the project because it only works on nominal data while our dataset consists of Numeric and Categorical Data. By using the KNeighborsClassifier for K-Nearest Neighbours, GaussianNB for Naïve Bayes, DecisionTreeClassifier for Decision Tree, SVC for Support Vector Machine, MLPClassifier for Neural network the models have been built with 5-fold cross-validation with GridSearchCV in every fold to obtain the best parameters for each fold. The parameters that have been tuned for each model is shown in Table 3.6.

Table 3.6: Parameters for Scikit-Learn Model

Classifier	Hyperparameters	Value
K Nearest Neighbours	n_neighbors	From 1.0 to 20
	metric	Euclidean, Manhattan, Minkowski
	weights	Uniform, Distance
Support Vector Machine	C	0.1,1,10,100
	gamma	1,0.1,0.01,0.001
	kernel	Rbf, Poly, Sigmoid
CART	max_features	Auto, Sqrt, Log2
	splitter	Best, Random
	min_samples_leaf	0.1,0.2,0.3
	min_samples_spilt	0.1,0.2,0.3,0.4,0.5
	max_depth	1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31
Naïve Bayes	-	-
Neural Network	solver	Sgd, Lbfgs, Adam
	hidden_layer_size	100,110,120,130,140,150
	activation	Identity, Logistic, Tanh, Relu
	learning_rate	Constant, Invscaling, Adaptive
	early_stopping	True, False

II. WEKA

The model that has been built using WEKA is C4.5 because there is no implementation of C4.5 available in the python libraries.

Table 3.7: Parameters for WEKA C4.5 Model

Classifier	Hyperparameters	Values
C4.5	M	1,2,3,4,5,6,7,8,9,10
	C	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9

5. Model Evaluation

The model will be evaluated by looking at its Balanced Accuracy, True Positive Rate and ROC/AUC Score. Due to the less amount of data collected for the project, for every model, 5-fold cross-validation is done to make sure all the rows in the dataset is fully utilised in training and testing. The main focus of the project is to predict the Failing Students, therefore the True Positive Rate of the Failing Classes which is 1, is prioritised in the evaluation. The model which gives the highest True Positive Rate in predicting the failing classes will be considered the best model for our project.

The ROC/AUC score is generated by using the roc_auc_score class in the Scikit-Learn Library. By using the classification_report class in the Scikit-Learn Library the classification report will also be generated to evaluate each of the model's performances by looking at the Balanced Accuracy, True Positive Rate. The Balanced Accuracy and the True Positive Rate are calculated automatically by Scikit-Learn based on the confusion matrix of the prediction done using the test set. Typically, a confusion matrix of a project will be *n* x *n* (*n* being the number of classes). For this project, the outcome is binary. Therefore, the confusion matrix will be 2 x 2. Table 3.8 below shows an example of a confusion matrix where the True Positive is failing students predicted, False positive is wrongly predicted failing students, True Negative is passing students predicted and false negative is wrongly predicted passing students.

Sample of confusion Matrix for the project:

Table 3.8: Sample of Binary Class Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 3.9: Confusion Matrix with Description

		Predicted	
		Positive (Fail)	Negative (Pass)
Actual	Positive (Fail)	Failing Students	Wrongly Predicted Passing Students
	Negative (Pass)	Wrongly Predicted Failing Students	Passing Students

True Positive Rate is also known as Recall is calculated using the ratio of true positives of a current class to the sum of its true positives and false negatives.

$$Recall\ of\ Fail = \frac{TP}{TP + FN} \tag{3.1}$$

$$= \frac{Failing\ Students}{Failing\ Students + Wrongly\ Predicted\ Passing\ Students}$$

True Negative Rate is also known as Specificity. It is calculated using the ratio of true negatives of a current class to the sum of its true negatives and false positives.

$$Specificity\ of\ Pass = \frac{TN}{TN + FP} \tag{3.2}$$

$$= \frac{Passing\ Students}{Passing\ Students + Wrongly\ Predicted\ Failing\ Students}$$

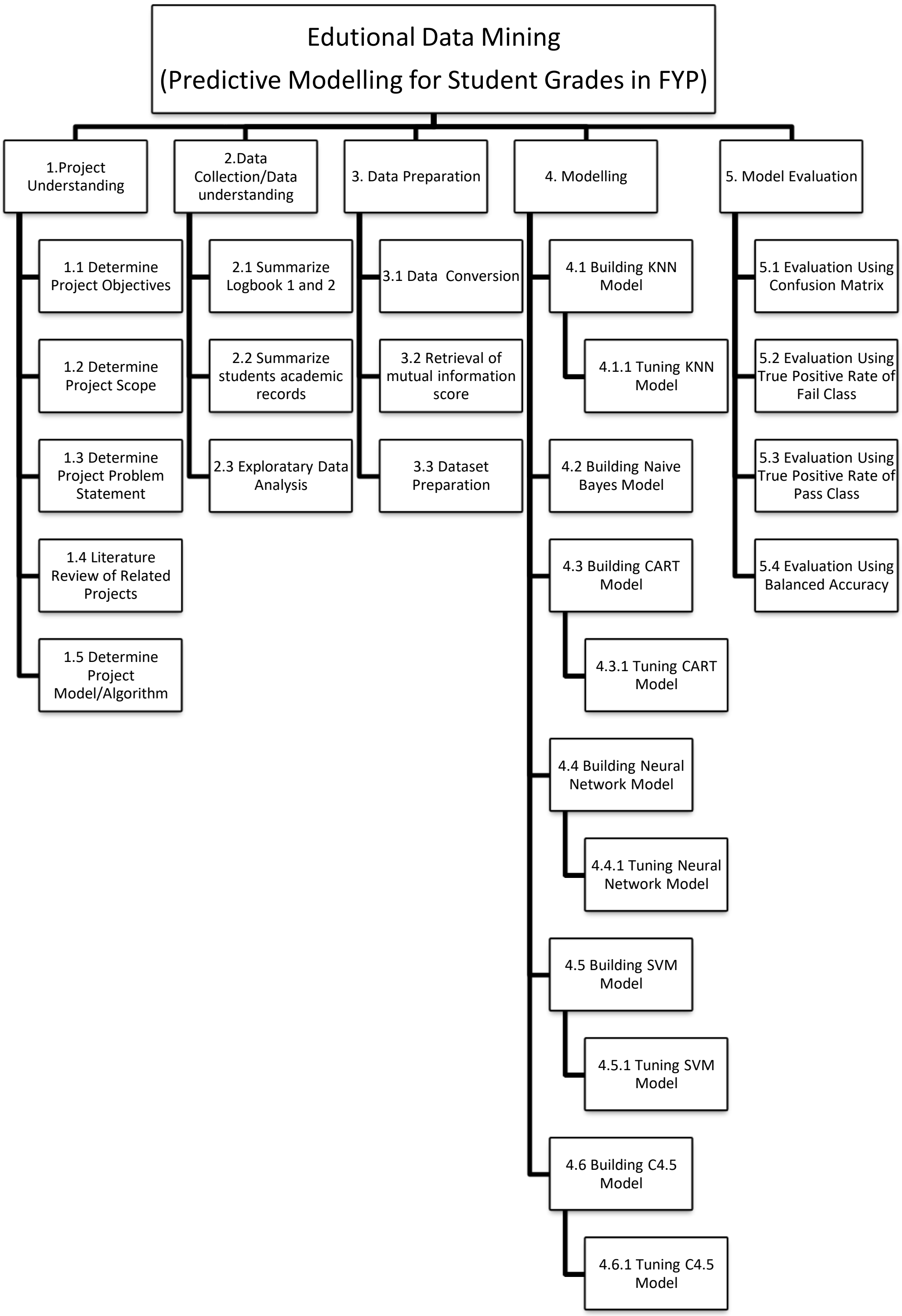
Balanced Accuracy is the average of Recall and Specificity:

$$Balanced\ Accuracy = \frac{Recall + Specificty}{2} \tag{3.3}$$

3.2 Work Breakdown Structure of the Project

Figure 3.9 shows the work breakdown structure of the project. The WBS is to be read from left to right starting from domain understanding and ending with model evaluation. The sub-task in each of the main task is to be read from top to bottom. The main task and its sub-task are to be finished before proceeding to the next main task.

Figure 3.9: Work Breakdown Structure of Project

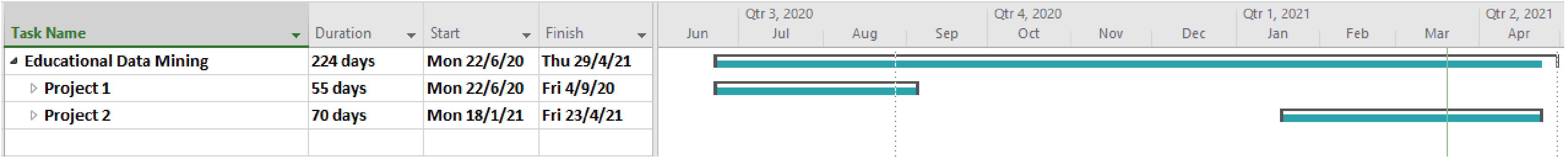

Edutional Data Mining
(Predictive Modelling for Student Grades in FYP)
1.Project Understanding
1.1 Determine Project Objectives
1.2 Determine Project Scope
1.3 Determine Project Problem Statement
1.4 Literature Review of Related Projects
1.5 Determine Project Model/Algorithm
2.Data Collection/Data understanding
2.1 Summarize Logbook 1 and 2
2.2 Summarize students academic records
2.3 Exploratary Data Analysis
3. Data Preparation
3.1 Data Conversion
3.2 Retrieval of mutual information score
3.3 Dataset Preparation
4. Modelling
4.1 Building KNN Model
4.1.1 Tuning KNN Model
4.2 Building Naive Bayes Model
4.3 Building CART Model
4.3.1 Tuning CART Model
4.4 Building Neural Network Model
4.4.1 Tuning Neural Network Model
4.5 Building SVM Model
4.5.1 Tuning SVM Model
4.6 Building C4.5 Model
4.6.1 Tuning C4.5 Model
5. Model Evaluation
5.1 Evaluation Using Confusion Matrix
5.2 Evaluation Using True Positive Rate of Fail Class
5.3 Evaluation Using True Positive Rate of Pass Class
5.4 Evaluation Using Balanced Accuracy

3.3 Gantt Chart of Project

This section shows the Gantt Chart for Project 1 and Project 2. All the planned task are made sure to abide the schedule planned in order to ease the project completion. The Overall Gantt Chart for Project 1 and Project 2 is shown in Figure 3.10, Figure 3.11 and Figure 3.12. The detailed Gantt Chart can be found in the Appendix A to Appendix G.

I. Overall Gantt Chart for the Whole Project

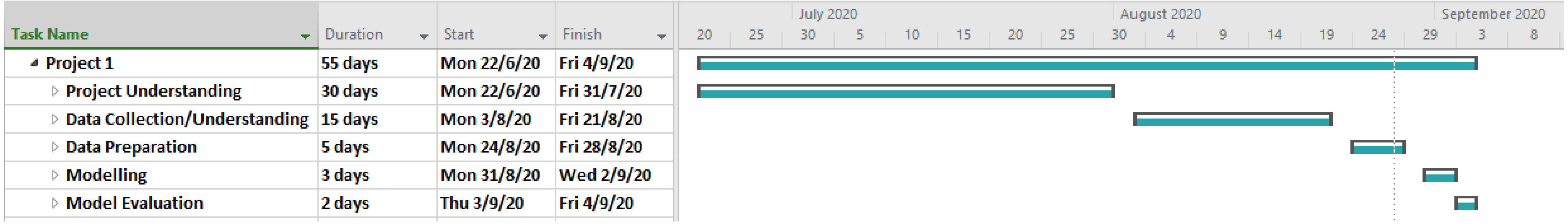
Figure 3.10: Overall Gantt Chart for the whole Project



Task Name	Duration	Start	Finish
◢ Educational Data Mining	224 days	Mon 22/6/20	Thu 29/4/21
▷ Project 1	55 days	Mon 22/6/20	Fri 4/9/20
▷ Project 2	70 days	Mon 18/1/21	Fri 23/4/21

II. Overall Gantt Chart for Project 1

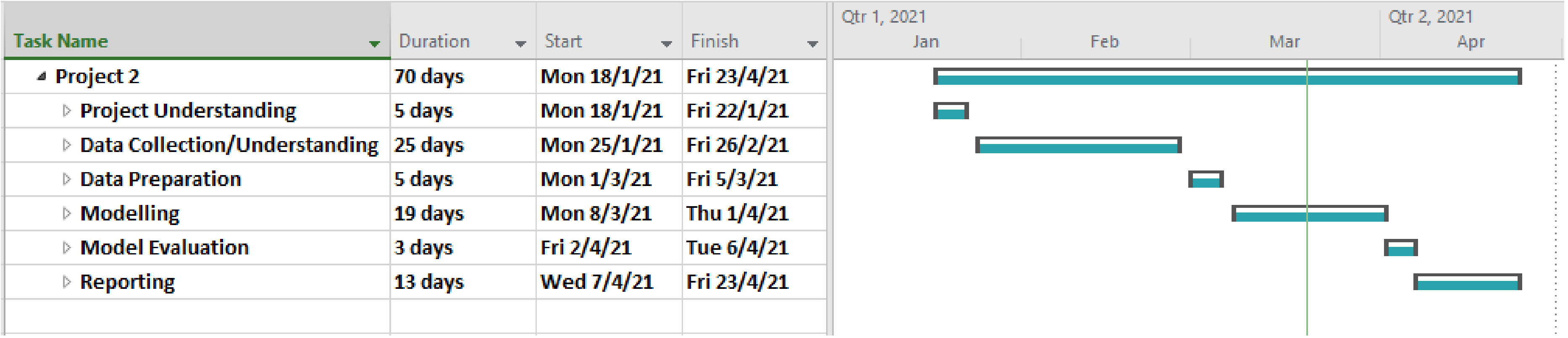
Figure 3.11: Overall Gantt Chart for Project 1



Task Name	Duration	Start	Finish
◢ Project 1	55 days	Mon 22/6/20	Fri 4/9/20
▷ Project Understanding	30 days	Mon 22/6/20	Fri 31/7/20
▷ Data Collection/Understanding	15 days	Mon 3/8/20	Fri 21/8/20
▷ Data Preparation	5 days	Mon 24/8/20	Fri 28/8/20
▷ Modelling	3 days	Mon 31/8/20	Wed 2/9/20
▷ Model Evaluation	2 days	Thu 3/9/20	Fri 4/9/20

III. Overall Gantt Chat for Project 2

Figure 3.12: Overall Gantt Chart for Project 2



Task Name	Duration	Start	Finish
Project 2	70 days	Mon 18/1/21	Fri 23/4/21
Project Understanding	5 days	Mon 18/1/21	Fri 22/1/21
Data Collection/Understanding	25 days	Mon 25/1/21	Fri 26/2/21
Data Preparation	5 days	Mon 1/3/21	Fri 5/3/21
Modelling	19 days	Mon 8/3/21	Thu 1/4/21
Model Evaluation	3 days	Fri 2/4/21	Tue 6/4/21
Reporting	13 days	Wed 7/4/21	Fri 23/4/21

CHAPTER 4

Results and Discussion

4.1 Introduction

The project aims to do predictions for students grades in FYP using the classifications model chosen. The chosen classification model for the project is K-Nearest Neighbours (KNN), Support Vector Machine (SVM), CART, C4.5, Naïve Bayes (NB) and Neural Network (NN). The models are built-in python language and Machine Learning Tools, WEKA. Python is a common language used for building classification models as its Scikit-Learn Library has built-in classes for various kinds of models and it can produce necessary output for model evaluation. These models will be evaluated based on the True Positive Rate, Balanced Accuracy and the ROC/AUC score with more focus on True Positive Rate, Recall.

4.2 Modelling and Result

This section of the report presents the result for each of the classification model chosen. K-Nearest Neighbours, Support Vector Machine, CART, C4.5, Naïve Bayes and Neural network are trained and tested with 5-fold cross-validation on all the 14-dataset created. These models are evaluated and compared based on the highest True Positive Rate (Recall) on different datasets. If 2 datasets have the same true positive rate, then the comparison will be done using True Negative Rate (Specificity), followed by balanced Accuracy and lastly, ROC/AUC Score.

4.2.1 K-Nearest Neighbours

Table 4.1: Results for K-Nearest Neighbours on the 14 datasets

Classifier	K-Nearest Neighbours			
Evaluation Metrics	Balanced Accuracy	Recall for Fail	Specificity for Pass	ROC/AUC Score
Dataset				
1	0.863	0.747	0.979	0.91
2	0.85	0.713	0.987	0.867
3	0.827	0.68	0.974	0.823
4	0.829	0.68	0.979	0.829
5	0.831	0.68	0.983	0.829
6	0.829	0.68	0.979	0.827
7	**0.865**	**0.747**	**0.983**	**0.867**
8	0.814	0.64	0.987	0.85
9	0.791	0.6	0.983	0.849
10	0.796	0.6	0.991	0.852
11	0.834	0.68	0.987	0.85
12	0.834	0.68	0.987	0.85
13	0.838	0.68	0.996	0.869
14	0.838	0.68	0.996	0.869

Table 4.2: Parameters for each fold on best result Dataset

Parameters Folds	n_neighbors	metric	weights
1	5	Euclidean	Uniform
2	5	Euclidean	Uniform
3	3	Euclidean	Uniform
4	1	Euclidean	Uniform
5	1	Euclidean	Uniform

Table 4.1 shows the result of K-Nearest Neighbours where the best result obtained is on Dataset 7 (Highlighted in Red). The highest Recall obtained on all the datasets is 0.747 on dataset 1 and 7. Dataset 7 have a higher specificity of 0.983 compared to dataset 1 with a specificity value of 0.979. Dataset 7 also have the highest balanced accuracy value of 0.865 compared to others dataset. It also obtained a decent value for ROC/AUC Score which is 0.867.

The parameter used to build each fold of the model on dataset 7 is shown in Table 4.2, where the majority of the fold uses “Euclidean” for the distance metric and “Uniform” for the weight function. It is observed that all 5 folds of the model use a low number of neighbours.

4.2.2 Support Vector Machine

Table 4.3: Results for Support Vector Machine on the 14 datasets

Classifier	Support Vector Machine			
Evaluation Metrics	Balanced Accuracy	Recall for Fail	Specificity for Pass	ROC/AUC Score
Dataset				
1	0.908	0.847	0.97	0.976
2	**0.913**	**0.847**	**0.979**	**0.95**
3	0.89	0.813	0.966	0.946
4	0.865	0.747	0.983	0.897
5	0.865	0.747	0.983	0.886
6	0.863	0.747	0.979	0.885
7	0.858	0.747	0.97	0.958
8	0.876	0.787	0.966	0.921
9	0.858	0.747	0.97	0.95
10	0.838	0.707	0.97	0.956
11	0.864	0.753	0.974	0.966
12	0.85	0.713	0.987	0.932
13	0.869	0.747	0.991	0.935
14	0.869	0.747	0.991	0.935

Table 4.4: Parameters for each fold on best result Dataset

Parameters Folds	C	gamma	kernel
1	0.1	1	Poly
2	0.1	1	Poly
3	0.1	1	Poly
4	0.1	1	Poly
5	0.1	1	Poly

Table 4.3 shows the results of the Support Vector Machine where the best results obtained is on Dataset 2 (Highlighted in Red). The highest recall value obtained is 0.847 on dataset 1 and 2. But dataset 2 is having a slightly higher Specificity value which is 0.979, compared to dataset 1, which is 0.97. Dataset 2 is concluded to be the dataset that brings out the best result on the Support Vector Machine. The Balanced Accuracy value of dataset 2 is the highest among all the dataset which is 0.913. Dataset 2 also obtained a decent ROC/AUC score which is 0.95.

The parameter used to build each fold of the support vector machine is shown in Table 4.4 with all folds using “0.1" for the C value, “1” for Gamma and “Poly” for the kernel.

4.2.3 CART

Table 4.5: Results for CART on the 14 datasets

Classifier	CART			
Evaluation Metrics	Balanced Accuracy	Recall for Fail	Specificity for Pass	ROC/AUC Score
Dataset				
1	0.724	0.487	0.962	0.746
2	0.715	0.447	0.983	0.749
3	0.5	0	1	0.7
4	0.654	0.32	0.987	0.7
5	0.733	0.487	0.979	0.794
6	0.733	0.487	0.979	0.732
7	0.729	0.48	0.979	0.754
8	0.598	0.2	0.996	0.763
9	0.636	0.28	0.991	0.625
10	0.654	0.32	0.987	0.746
11	0.5	0	1	0.662
12	0.5	0	1	0.672
13	0.694	0.4	0.987	0.803
14	**0.748**	**0.527**	**0.97**	**0.851**

Table 4.6: Parameters for each fold on best result Dataset

Parameters / Folds	Max_features	splitter	Min_samples_leaf	Min_samples_split	Max_depth
1	Log2	Best	0.1	0.3	16
2	Auto	Best	0.1	0.1	4
3	Auto	Best	0.1	0.5	1
4	Log2	Best	0.1	0.1	25
5	Sqrt	Best	0.1	0.1	7

Table 4.5 shows the results for the CART decision tree, where the best recall value is 0.527 on dataset 14. The CART decision tree performs very badly as all the dataset have recall value lower than 0.5 except for one on dataset 14. Although CART performs badly in predicting failing student, it can still perform very well in predicting the passing students. Dataset 14 has a specificity of 0.97, Balanced Accuracy of 0.748 and a ROC/AUC score of 0.851. All of the datasets have a specificity above 0.96 where some of the datasets have a specificity value of 1.

The parameters used to build each fold of the CART decision tree is also shown where the max_features parameter is either “Log2” or “Auto”. Splitter and min_samples_leaf have the same value for every fold which is “Best” and “0.1” respectively. The min_samples_spilt consist of 0.3 in fold 1, 0.5 on fold 3 and 0.1 on fold 2,4,5. Lastly, it is observed that no similarities or pattern can be found for the max_depth parameter on every fold.

4.2.4 C4.5

Table 4.7: Results for C4.5 on the 14 datasets

Classifier	C4.5			
Evaluation Metrics	Balanced Accuracy	Recall for Fail	Specificity for Pass	ROC/AUC Score
Dataset				
1	0.852	0.704	1	0.88
2	0.864	0.741	0.987	0.871
3	**0.8825**	**0.778**	**0.987**	**0.858**
4	0.852	0.704	1	0.847
5	0.85	0.704	0.996	0.843
6	0.85	0.704	0.996	0.843
7	0.852	0.704	1	0.886
8	0.852	0.704	1	0.886
9	0.852	0.704	1	0.885
10	0.85	0.704	0.996	0.85
11	0.85	0.704	0.996	0.85
12	0.848	0.704	0.992	0.847
13	0.848	0.704	0.992	0.847
14	0.848	0.704	0.992	0.847

Table 4.8: Parameters for each fold on best result Dataset

Parameters Folds	M	C
1	2	0.6
2	2	0.6
3	2	0.6
4	2	0.6
5	2	0.6

Table 4.7 shows the result of the C4.5 decision tree. It is the only decision tree that is build using WEKA in our project. From the table, it is observed that the C4.5 Decision Tree perform best on Dataset 3 with a recall of 0.778, Specificity of 0.987, Balanced Accuracy of

0.8825 and ROC/AUC score of 0.858. The best parameters for each fold are M with the value of 2 and C with the value of 0.6, which is shown in table 4.8.

4.2.5 Naïve Bayes

Table 4.9: Results for Naïve Bayes on the 14 datasets

Classifier	Naïve Bayes			
Evaluation Metrics	Balanced Accuracy	Recall for Fail	Specificity for Pass	ROC/AUC Score
Dataset				
1	0.863	0.747	0.979	0.976
2	0.861	0.747	0.974	0.979
3	0.856	0.747	0.966	0.971
4	0.872	0.787	0.958	0.973
5	0.868	0.787	0.949	0.965
6	0.759	0.727	0.792	0.873
7	0.759	0.727	0.792	0.854
8	0.737	0.687	0.788	0.849
9	0.764	0.727	0.801	0.842
10	0.764	0.727	0.801	0.83
11	0.764	0.727	0.801	0.822
12	0.705	0.72	0.69	0.764
13	0.725	0.76	0.69	0.763
14	**0.617**	**0.793**	**0.441**	**0.767**

Table 4.9 shows the result of Naïve Bayes on different datasets. It can be observed that Naïve Bayes has the best recall on Dataset 14 but its specificity value is lowest on dataset 14 which is 0.441.

Looking at other results, dataset 4 have the most balanced result with a recall value of 0.787, which is slightly lower than dataset 14. But it has a specificity value of 0.958, which is 0.517 more than dataset 14. The balanced accuracy score of datasets 4 is also the highest among all of the datasets, with a value of 0.872. Due to the objectives of the project, dataset 14 is still chosen as the best performing dataset because of the higher recall value, but it is also observed that Naïve Bayes also perform fairly well with others dataset. The GaussianNB class in Scikit-Learn for the Naïve Bayes model also has no parameters for tuning.

4.2.6 Neural Network

Table 4.10: Results for Neural Network on the 14 datasets

Classifier	Neural Network			
Evaluation Metrics	Balanced Accuracy	Recall for Fail	Specificity for Pass	ROC/AUC Score
Dataset				
1	**0.576**	**0.76**	**0.391**	**0.577**
2	0.631	0.48	0.783	0.632
3	0.662	0.38	0.945	0.701
4	0.649	0.507	0.791	0.519
5	0.838	0.713	0.962	0.83
6	0.657	0.36	0.954	0.739
7	0.574	0.187	0.962	0.637
8	0.819	0.68	0.957	0.765
9	0.834	0.707	0.962	0.853
10	0.639	0.32	0.958	0.547
11	0.794	0.627	0.962	0.818
12	0.775	0.58	0.97	0.722
13	0.732	0.52	0.945	0.803
14	0.76	0.553	0.966	0.822

Table 4.11: Parameters for each fold on best result Dataset

Parameters Folds	solver	Hidden_layer_size	activation	Learning_rate	Early_stopping
1	Adam	100	Tanh	Constant	True
2	Sgd	110	Logistic	Invscaling	True
3	Sgd	110	Logistic	Invscaling	True
4	Adam	140	Relu	Constant	True
5	Sgd	130	Relu	Invscaling	True

Table 4.10 shows the result for the Neural network on all the datasets. It can be observed that Neural Network perform best on dataset based on recall value. The recall value of 0.76 on dataset 1 indicates that the neural network only needs 1 attribute to do predictions on failing students. However, with only 1 attribute, Neural Network performs badly on predicting passing students with a specificity of 0.391. On Datasets 1 Neural network also have a ROC/AUC score of 0.577 and Balanced Accuracy of 0.576.

Due to the objectives of the project, Dataset 1 is chosen to be the best dataset for the Neural network to predict failing students. If an overall performance where passing and failing students' predictions are taken into consideration Dataset 5 will be the best dataset for Neural network with a balanced accuracy of 0.838, recall of 0.713 and specificity of 0.962.

4.3 Experiment Summary

Table 4.12: Summary of Best Result in predicting failing students

Model	Recall for Fail	Specificity for Pass	Balanced Accuracy	ROC/AUC Score	Dataset
KNN	0.747	0.983	0.865	0.867	7
SVM	**0.847**	**0.979**	**0.913**	**0.95**	**2**
CART	0.527	0.97	0.748	0.851	14
C4.5	0.778	0.987	0.8825	0.858	3
NB	0.793	0.441	0.617	0.767	14
NN	0.760	0.391	0.576	0.577	1

Based on the result summary, it is shown that even though Neural Network and Naïve Bayes has acquired a high value of recall and is good at predicting failing students, it does not mean that they are also good in predicting passing students because they have scored low specificity values which are 0.391 and 0.441, respectively. If the True positive rate of both failing and passing students is considered, a model dataset with higher balanced accuracy should be chosen as the best performing model. Due to the reason that the objective of this project is to choose the model which performs best in predicting failing students hence the high level of recall value if prioritised.

Based on the result summary, it is observed that the worst model for predicting failing student using the provided data set is the CART model. CART model obtains the highest recall value of 0.527. It also does not perform well as other dataset's recall value are less than 0.5. The dataset that obtains the highest recall value is also dataset 14 where most attributes are needed. Comparing to other models such as Neural Network, C4.5, SVM and KNN, a much better result can be obtained by using lesser attributes.

On the contrary, the best performing model in predicting failing student is SVM. SVM has the highest recall among all the models, with a value of 0.847. SVM also requires very few features to do prediction. This is because the best result for SVM is achieved using dataset 2 with only 2 features, namely "Total Logs Submission in Logbook 2" and "Total Number of Retakes".

Table 4.13: Summary of Best Result in Predicting both classes

Model	Recall for Fail	Specificity for Pass	Balanced Accuracy	ROC/AUC Score	Datasets
KNN	0.747	0.983	0.865	0.867	7
SVM	**0.847**	**0.979**	**0.913**	**0.95**	**2**
CART	0.527	0.97	0.748	0.851	14
C4.5	0.778	0.987	0.8825	0.858	3
NB	0.787	0.958	0.872	0.973	4
NN	0.713	0.962	0.838	0.83	5

By considering both classes True Positive rate table 4.13 shows the best summary of the result for each of the model. If a small compromise of 0.006 and 0.047 on Naïve Bayes and Neural Network recall value is made, it can increase the Specificity for both the model by 0.517 and 0.571 respectively. This way the models can perform well on predictions of both classes.

It is also observed that the majority of the Model that performs well only need a few attributes to do predictions. KNN performs wells with Datasets 7 with 7 feature, SVM performs wells with Dataset 2 with 2 features, C4.5 perform well with Dataset 3 with 3 features, Naïve Bayes perform well with Dataset 4 with 4 features and lastly Neural Network perform well with Dataset 5 with 5 features.

Furthermore, the predictive models that obtain satisfactory results in predicting failing students used features other than CGPA in the dataset. The result of CART, the only predictive model that uses all features to make predictions, is 0.527. Through empirical studies, it is found that there are at least 2 important features for the prediction of failing students, namely, Total Logs Submission in Logbook 2 and Total Number of Retakes. As all features are used for predictions and can bring out a satisfactory result, all features collected in this study are useful in predicting failing students.

Table 4.14 shows a summary of the predictive model that obtains satisfactory result in predictions and its features used.

Table 4.14: Summary of Features in Dataset

Classifier	Dataset	Features
KNN	7	Total Logs Submission in Logbook 2, Total Number of Retakes, Total Number of Subjects Failed, No of Trimesters Before FYP2, No of Subjects taken with Project II, Total Number of Listing, Total Logs Submission in Logbook 1
SVM	2	Total Logs Submission in Logbook 2, Total Number of Retakes
C4.5	3	Total Logs Submission in Logbook 2, Total Number of Retakes, Total Number of Subjects Failed
NB	4	Total Logs Submission in Logbook 2, Total Number of Retakes, Total Number of Subjects Failed, No of Trimesters Before FYP2
NN	5	Total Logs Submission in Logbook 2, Total Number of Retakes, Total Number of Subjects Failed, No of Trimesters Before FYP2, No of Subjects taken with Project II

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Machine learning's success in predicting failing students relies on the good use of data and predictive models. It is important to choose the right predictive models for the collected dataset to achieve the optimum result.

Objective 1 of the project is to create a dataset with attributes that can predict students' grades. This objective is achieved on the steps of data collections. Through summarising the students' academic record and Logbook, a raw dataset with 263 columns and 15 attributes is created.

Objective 2 of the project is to identify the useful attributes in a dataset for predictions. This objective is achieved by using the mutual_info_classif class of python's Scikit-Learn Library. Based on mutual information, 14 datasets are created for predictions where all datasets consist of different attributes. Through empirical studies, it is found that there are features that are useful in predicting student failure other than CGPA. It is found that all the 14 features collected is useful for predicting failing students.

Objectives 3 of the project is to predict the grades of students taking FYP using K-Nearest neighbours, CART, C4.5, Naïve Bayes, Support Vector Machine and Neural network. This objective is achieved during modelling when cross-validation is used. Cross-validation uses different parts of the dataset to do prediction on every iteration.

The project's last objective is to select the best predictive model using True Positive Rate in Prediction of Failing Student. This objective is achieved at the end of the experiment, where SVM is concluded to be the best predictive model. It has the highest recall among all the models, which is 0.847, a specificity as high as 0.979, a balanced accuracy of 0.913 and a ROC/AUC score of 0.95. It also performs well on dataset 2, which only uses 2 attributes which is Total Logs Submission in Logbook 2 and Total Number of Retakes.

5.2 Recommendations for future work

5.2.1 Train Model with larger datasets

The current dataset only consists of 263 rows where only 27 rows are the sample for failing students. To enhance the performance of the model more data are required. Training with more dataset allows the model to learn more patterns of failing students and can greatly increase the recall and balanced accuracy of the models.

5.2.2 Use more Predictive Model for Training

In this project, only seven classification algorithms are used: K-Nearest Neighbours, CART, C4.5, Naïve Bayes, Support Vector Machine and Neural Network. Many different classification models such as Logistic Regression, Random Forest, Stochastic Gradient Descent, Linear Regression, etc. can be used to achieve the objectives of this project. Different models might acquire different results, other models might produce a better result with the same dataset.

5.2.3 Increase Attributes Types

The attributes in this project covered mostly on the student's academic performance and logbook status. As observed during data understanding steps, there are outliers. Students that have high CGPA might also fail in their FYP2. Therefore, there might be more factors that will help in predicting a student's failure in Final Year Project. Examples of attributes that can be collected are students' household income, students' parental education level, students' part-time job status, students' relationship status. etc. Different attributes might increase the accuracy of the model as these attributes might have a relation to why a student will fail their FYP.

REFERENCES

Alazzam, H., Sharieh, A. and Sabri, K.E., 2020. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. *Expert systems with applications*, *148*, p.113249.

Almahadeen, L., Akkaya, M. and Sari, A., 2017. Mining student data using CRISP-DM model. *International Journal of Computer Science and Information Security*, *15*(2), p.305.

Al-Shehri, H., Al-Qarni, A., Al-Saati, L., Batoaq, A., Badukhen, H., Alrashed, S., Alhiyafi, J. and Olatunji, S.O., 2017, April. Student performance prediction using support vector machine and k-nearest neighbor. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)* (pp. 1-4). IEEE.

Asif, R., Merceron, A., Ali, S.A. and Haider, N.G., 2017. Analyzing undergraduate students' performance using educational data mining. *Computers & Education*, *113*, pp.177-194

Athani, S.S., Kodli, S.A., Banavasi, M.N. and Hiremath, P.S., 2017, July. Student performance predictor using multiclass support vector classification algorithm. In *2017 International Conference on Signal Processing and Communication (ICSPC)* (pp. 341-346). IEEE.

Bahadir, E., 2016. Using Neural Network and Logistic Regression Analysis to Predict Prospective Mathematics Teachers' Academic Success upon Entering Graduate Education. *Educational Sciences: Theory and Practice*, *16*(3), pp.943-964.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. and Lopez, A., 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, pp.189-215.

Cheng, X., 2000. Asian students' reticence revisited. *System*, *28*(3), pp.435-446.

Clark, A., 2018. The Machine Learning Audit—CRISP-DM Framework. *ISACA Journal, 1*.

de Almeida Lima, M.N.C., Alves, G.O., Soares, W.L. and de Araújo Fagundes, R.A., 2019. Educational Data Mining: A Hybrid Approach to Predicting Academic Performance of Students. In *MLDM (2)* (pp. 500-514).

Etter, E.R., Burmeister, S.L. and Elder, R.J., 2000. Improving student performance and retention via supplemental instruction. *Journal of Accounting Education*, *18*(4), pp.355-368.

Huang, S., Cai, N., Pacheco, P.P., Narrandes, S., Wang, Y. and Xu, W., 2018. Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics-Proteomics*, *15*(1), pp.41-51.

Kabakchieva, D., 2012. Student performance prediction by using data mining classification algorithms. *International journal of computer science and management research*, *1*(4), pp.686-690.

Kotsiantis, S.B., Zaharakis, I. and Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), pp.3-24.

Langley, P., 1994, November. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall symposium on relevance* (Vol. 184, pp. 245-271).

Latham, P.E. and Roudi, Y., 2009. Mutual information. *Scholarpedia*, *4*(1), p.1658.

Lenin, T. and Chandrasekaran, N., Students' Performance Prediction Modelling using Classification Technique in R.

Marbán, O., Segovia, J., Menasalvas, E. and Fernández-Baizán, C., 2009. Toward data mining engineering: A software engineering approach. *Information systems*, *34*(1), pp.87-107.

Mijwel, M.M., 2018. Artificial neural networks advantages and disadvantages. *Retrieved from LinkedIn: https://www. linkedin. com/pulse/artificial-neuralnet works-advantages-disadvantages-maad-m-mijwel*.

Mohankumar, M., Amuthakkani, S. and Jeyamala, G., 2016. Comparative analysis of decision tree algorithms for the prediction of eligibility of a man for availing bank loan. *Age*, *19*, p.60.

Mueen, A., Zafar, B. and Manzoor, U., 2016. Modeling and predicting students' academic performance using data mining techniques. *International Journal of Modern Education and Computer Science*, *8*(11), p.36.

Naicker, N., Adeliyi, T. and Wing, J., 2020. Linear Support Vector Machines for Prediction of Student Performance in School-Based Education. *Mathematical Problems in Engineering*, *2020*.

Ramesh, V.A.M.A.N.A.N., Parkavi, P. and Ramar, K., 2013. Predicting student performance: a statistical and data mining approach. *International journal of computer applications*, *63*(8).

Saheed, Y.K., Oladele, T.O., Akanni, A.O. and Ibrahim, W.M., 2018. Student performance prediction based on data mining classification techniques. *Nigerian Journal of Technology*, *37*(4), pp.1087-1091.

Shahiri, A.M. and Husain, W., 2015. A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, *72*, pp.414-422.

Sharma, H. and Kumar, S., 2016. A survey on decision tree algorithms of classification in data mining. *International Journal of Science and Research (IJSR)*, *5*(4), pp.2094-2097.

Singh, A. and Lakshmiganthan, R., 2018. Impact of different data types on classifier performance of random forest, naive bayes, and k-nearest neighbors algorithms.

Yaacob, W.F.W., Nasir, S.A.M., Yaacob, W.F.W. and Sobri, N.M., 2019. Supervised data mining approach for predicting student performance. *Indones. J. Electr. Eng. Comput. Sci*, *16*, pp.1584-1592

Yu, L. and Liu, H., 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 856-863).

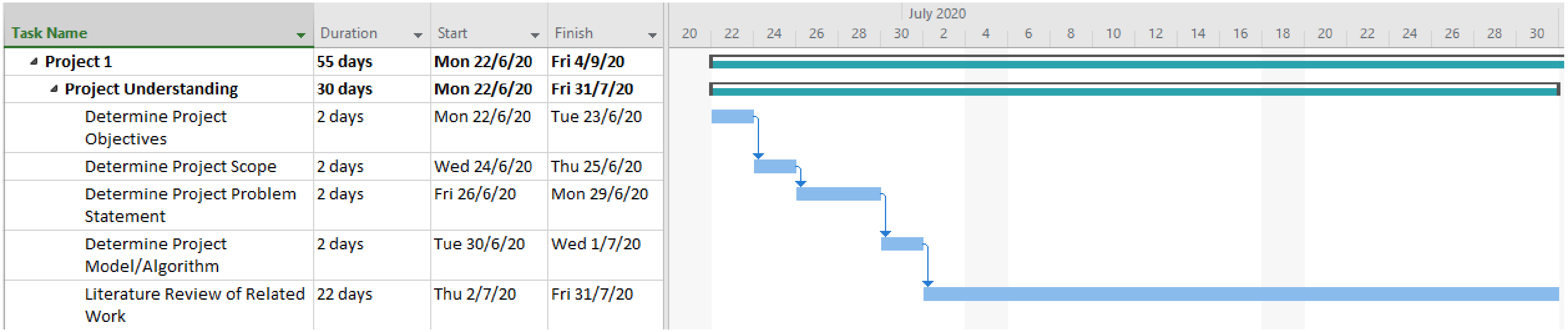
Zacharis, N.Z., 2016. Predicting student academic performance in blended learning using Artificial Neural Networks. *International Journal of Artificial Intelligence and Applications*, *7*(5), pp.17-29.

Zhang, S., Zhang, C. and Yang, Q., 2003. Data preparation for data mining. *Applied artificial intelligence*, *17*(5-6), pp.375-381.

APPENDICES

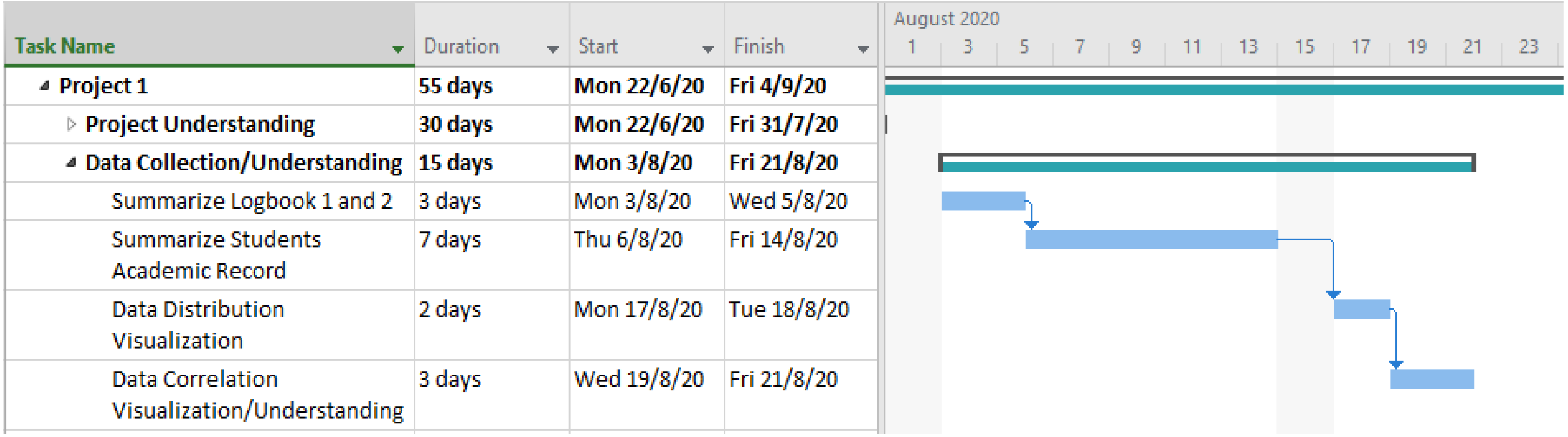
APPENDIX A: Detailed Gantt Chart for Project 1

I. Gantt Chart for Project 1-Project Understanding



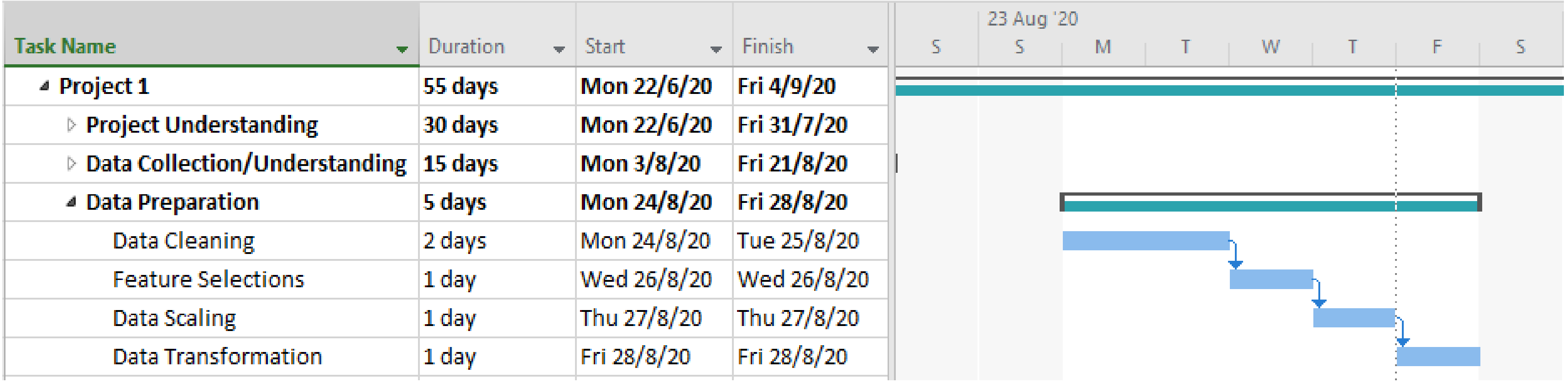
Task Name	Duration	Start	Finish
Project 1	**55 days**	**Mon 22/6/20**	**Fri 4/9/20**
Project Understanding	**30 days**	**Mon 22/6/20**	**Fri 31/7/20**
Determine Project Objectives	2 days	Mon 22/6/20	Tue 23/6/20
Determine Project Scope	2 days	Wed 24/6/20	Thu 25/6/20
Determine Project Problem Statement	2 days	Fri 26/6/20	Mon 29/6/20
Determine Project Model/Algorithm	2 days	Tue 30/6/20	Wed 1/7/20
Literature Review of Related Work	22 days	Thu 2/7/20	Fri 31/7/20

II. Gantt Chart for Project 1-Data Collection/Understanding



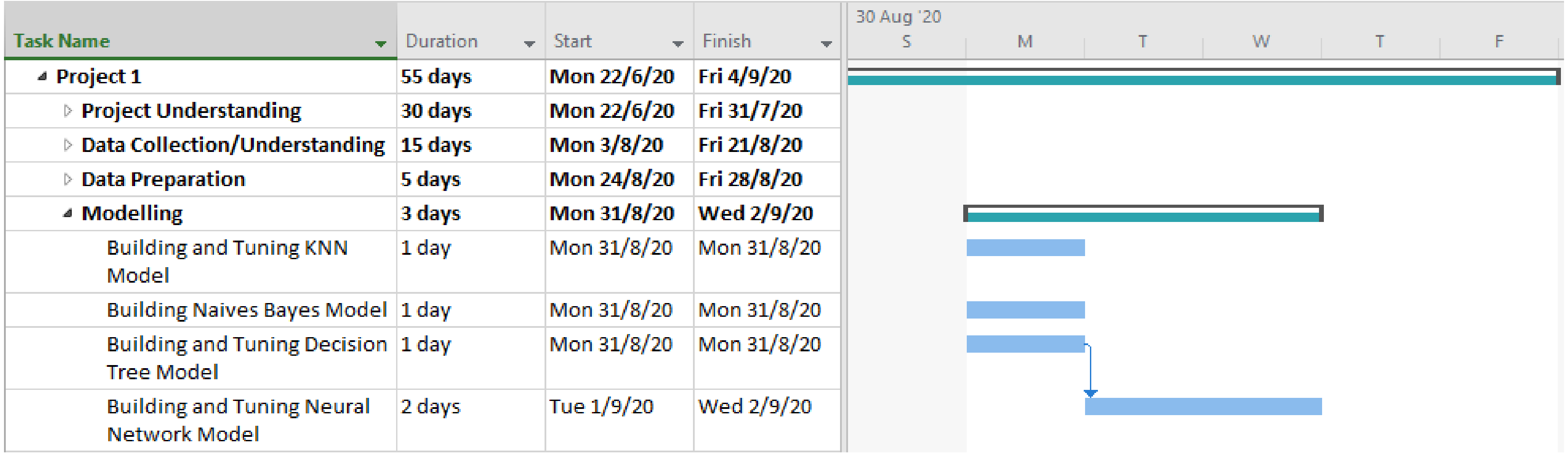
Task Name	Duration	Start	Finish
Project 1	**55 days**	**Mon 22/6/20**	**Fri 4/9/20**
Project Understanding	**30 days**	**Mon 22/6/20**	**Fri 31/7/20**
Data Collection/Understanding	**15 days**	**Mon 3/8/20**	**Fri 21/8/20**
Summarize Logbook 1 and 2	3 days	Mon 3/8/20	Wed 5/8/20
Summarize Students Academic Record	7 days	Thu 6/8/20	Fri 14/8/20
Data Distribution Visualization	2 days	Mon 17/8/20	Tue 18/8/20
Data Correlation Visualization/Understanding	3 days	Wed 19/8/20	Fri 21/8/20

III. Gantt Chart for Project 1-Data Preparation



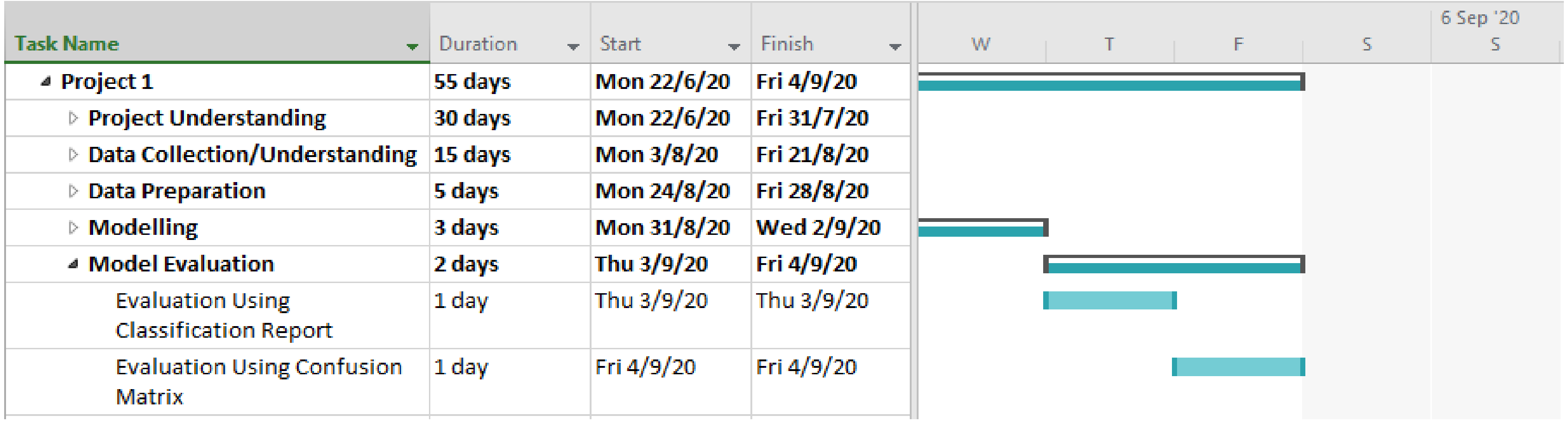
Task Name	Duration	Start	Finish
Project 1	**55 days**	**Mon 22/6/20**	**Fri 4/9/20**
Project Understanding	**30 days**	**Mon 22/6/20**	**Fri 31/7/20**
Data Collection/Understanding	**15 days**	**Mon 3/8/20**	**Fri 21/8/20**
Data Preparation	**5 days**	**Mon 24/8/20**	**Fri 28/8/20**
Data Cleaning	2 days	Mon 24/8/20	Tue 25/8/20
Feature Selections	1 day	Wed 26/8/20	Wed 26/8/20
Data Scaling	1 day	Thu 27/8/20	Thu 27/8/20
Data Transformation	1 day	Fri 28/8/20	Fri 28/8/20

IV. Gantt Chart for Project 1-Modelling



Task Name	Duration	Start	Finish
Project 1	**55 days**	**Mon 22/6/20**	**Fri 4/9/20**
Project Understanding	**30 days**	**Mon 22/6/20**	**Fri 31/7/20**
Data Collection/Understanding	**15 days**	**Mon 3/8/20**	**Fri 21/8/20**
Data Preparation	**5 days**	**Mon 24/8/20**	**Fri 28/8/20**
Modelling	**3 days**	**Mon 31/8/20**	**Wed 2/9/20**
Building and Tuning KNN Model	1 day	Mon 31/8/20	Mon 31/8/20
Building Naives Bayes Model	1 day	Mon 31/8/20	Mon 31/8/20
Building and Tuning Decision Tree Model	1 day	Mon 31/8/20	Mon 31/8/20
Building and Tuning Neural Network Model	2 days	Tue 1/9/20	Wed 2/9/20

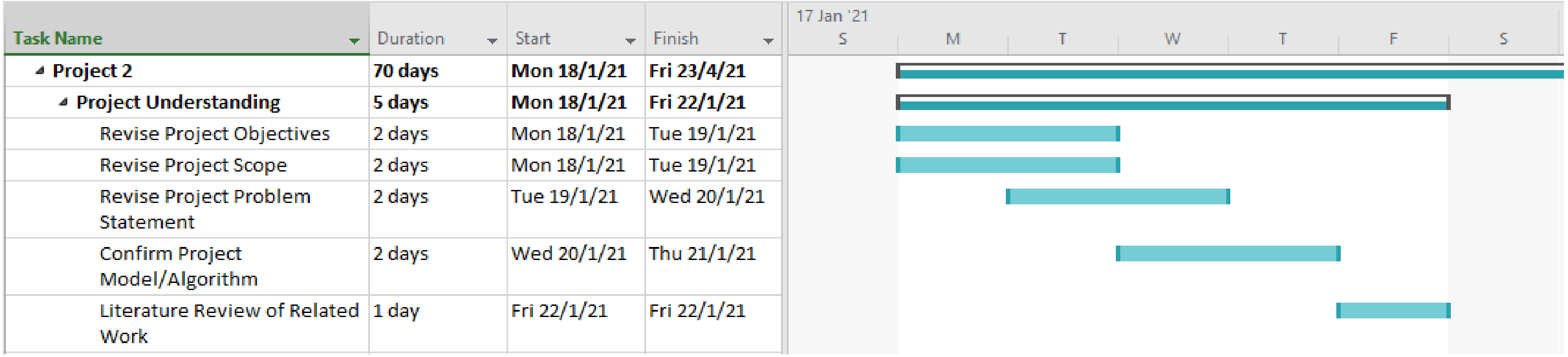
V. Gantt Chart for Project 1-Model Evaluation



Task Name	Duration	Start	Finish
Project 1	**55 days**	**Mon 22/6/20**	**Fri 4/9/20**
Project Understanding	**30 days**	**Mon 22/6/20**	**Fri 31/7/20**
Data Collection/Understanding	**15 days**	**Mon 3/8/20**	**Fri 21/8/20**
Data Preparation	**5 days**	**Mon 24/8/20**	**Fri 28/8/20**
Modelling	**3 days**	**Mon 31/8/20**	**Wed 2/9/20**
Model Evaluation	**2 days**	**Thu 3/9/20**	**Fri 4/9/20**
Evaluation Using Classification Report	1 day	Thu 3/9/20	Thu 3/9/20
Evaluation Using Confusion Matrix	1 day	Fri 4/9/20	Fri 4/9/20

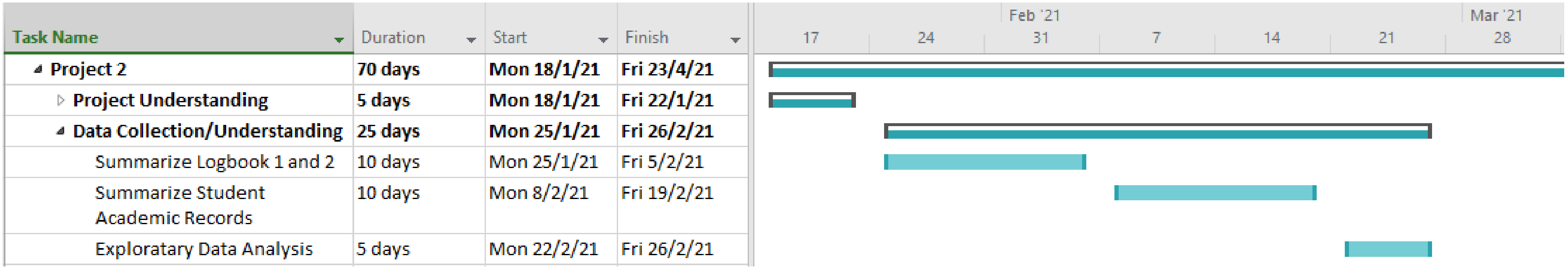
APPENDIX B: Detailed Gantt Chart for Project 2

I. Gantt Chart for Project 2- Project Understanding



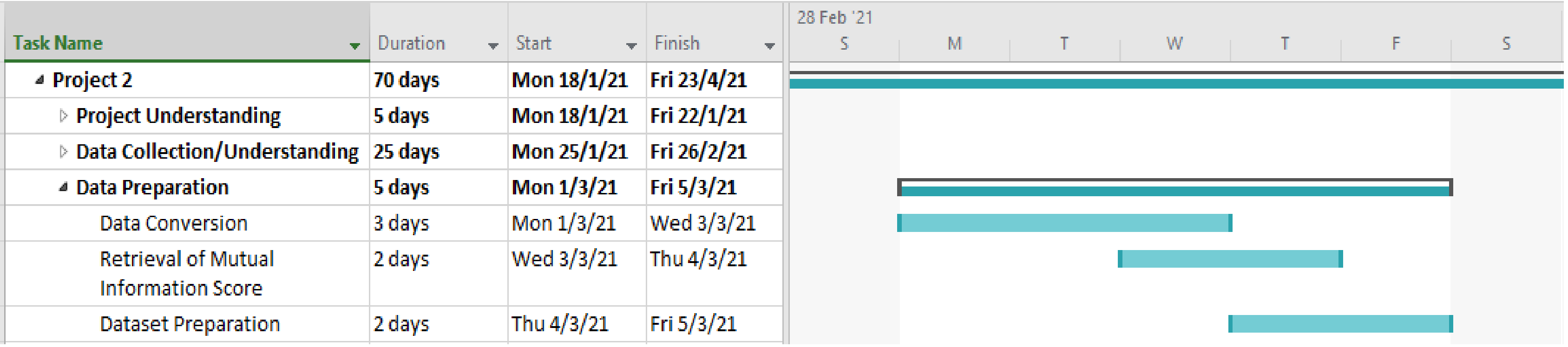
Task Name	Duration	Start	Finish
Project 2	**70 days**	**Mon 18/1/21**	**Fri 23/4/21**
Project Understanding	**5 days**	**Mon 18/1/21**	**Fri 22/1/21**
Revise Project Objectives	2 days	Mon 18/1/21	Tue 19/1/21
Revise Project Scope	2 days	Mon 18/1/21	Tue 19/1/21
Revise Project Problem Statement	2 days	Tue 19/1/21	Wed 20/1/21
Confirm Project Model/Algorithm	2 days	Wed 20/1/21	Thu 21/1/21
Literature Review of Related Work	1 day	Fri 22/1/21	Fri 22/1/21

II. Gantt Chart for Project 2-Data Collection/Understanding



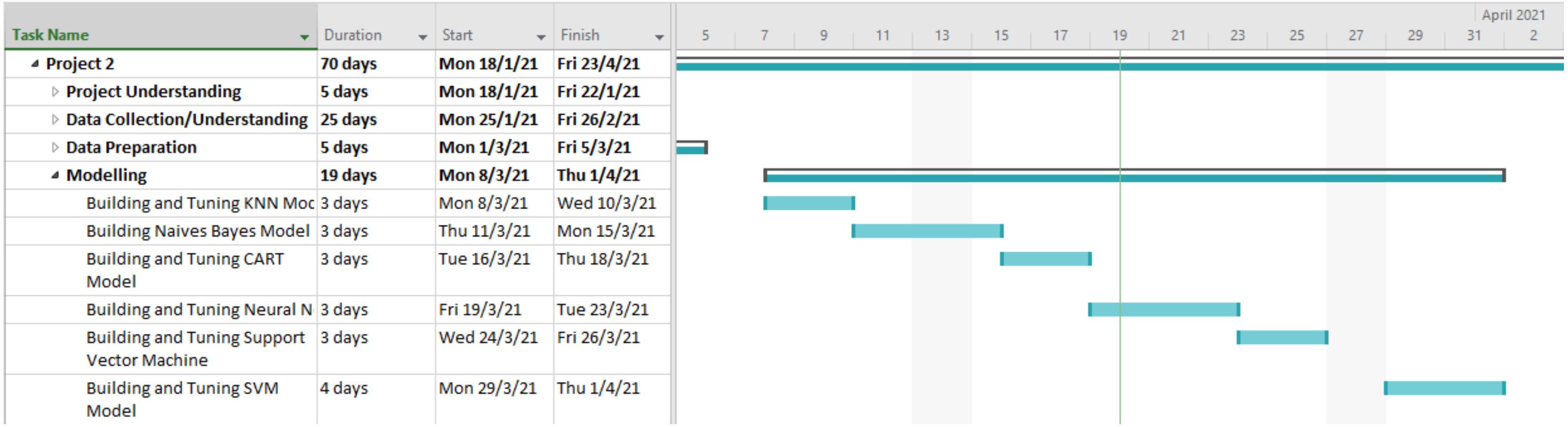
Task Name	Duration	Start	Finish
Project 2	**70 days**	**Mon 18/1/21**	**Fri 23/4/21**
Project Understanding	**5 days**	**Mon 18/1/21**	**Fri 22/1/21**
Data Collection/Understanding	**25 days**	**Mon 25/1/21**	**Fri 26/2/21**
Summarize Logbook 1 and 2	10 days	Mon 25/1/21	Fri 5/2/21
Summarize Student Academic Records	10 days	Mon 8/2/21	Fri 19/2/21
Exploratary Data Analysis	5 days	Mon 22/2/21	Fri 26/2/21

III. Gantt Chart for Project 2-Data Preparation



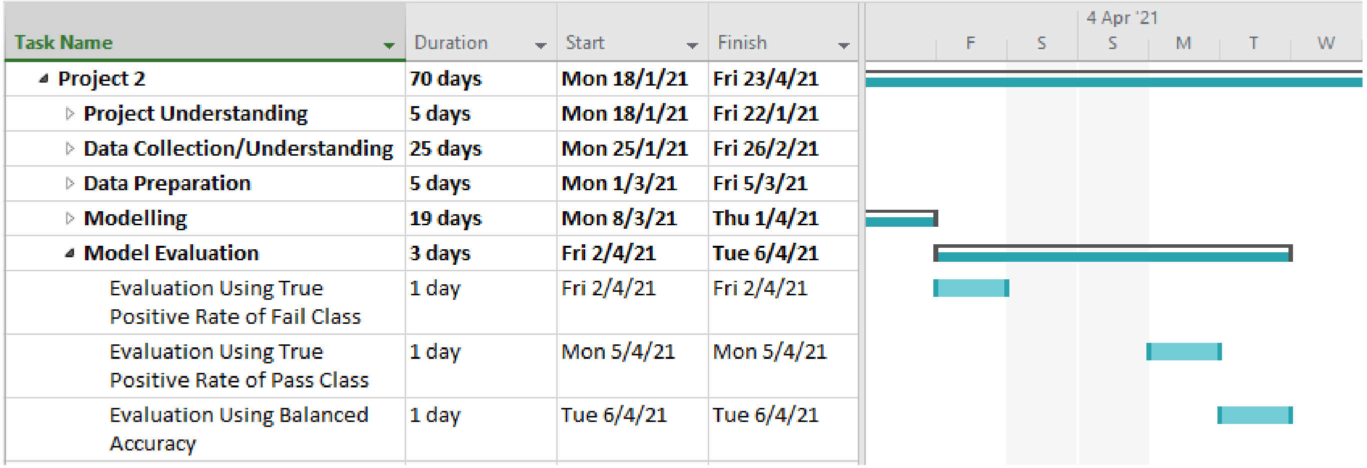
Task Name	Duration	Start	Finish
Project 2	**70 days**	**Mon 18/1/21**	**Fri 23/4/21**
Project Understanding	**5 days**	**Mon 18/1/21**	**Fri 22/1/21**
Data Collection/Understanding	**25 days**	**Mon 25/1/21**	**Fri 26/2/21**
Data Preparation	**5 days**	**Mon 1/3/21**	**Fri 5/3/21**
Data Conversion	3 days	Mon 1/3/21	Wed 3/3/21
Retrieval of Mutual Information Score	2 days	Wed 3/3/21	Thu 4/3/21
Dataset Preparation	2 days	Thu 4/3/21	Fri 5/3/21

IV. Gantt Chart for Project 2-Modelling



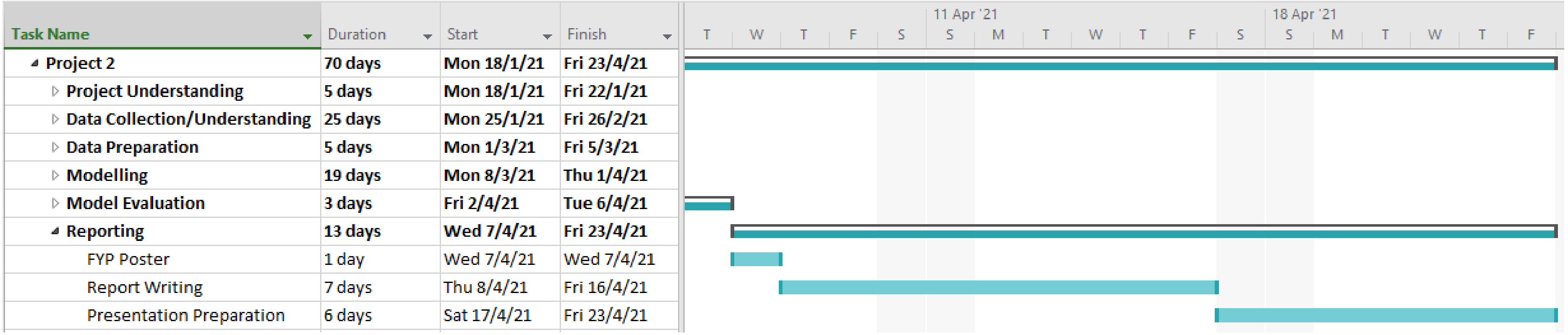
Task Name	Duration	Start	Finish
Project 2	**70 days**	**Mon 18/1/21**	**Fri 23/4/21**
Project Understanding	**5 days**	**Mon 18/1/21**	**Fri 22/1/21**
Data Collection/Understanding	**25 days**	**Mon 25/1/21**	**Fri 26/2/21**
Data Preparation	**5 days**	**Mon 1/3/21**	**Fri 5/3/21**
Modelling	**19 days**	**Mon 8/3/21**	**Thu 1/4/21**
Building and Tuning KNN Moc	3 days	Mon 8/3/21	Wed 10/3/21
Building Naives Bayes Model	3 days	Thu 11/3/21	Mon 15/3/21
Building and Tuning CART Model	3 days	Tue 16/3/21	Thu 18/3/21
Building and Tuning Neural N	3 days	Fri 19/3/21	Tue 23/3/21
Building and Tuning Support Vector Machine	3 days	Wed 24/3/21	Fri 26/3/21
Building and Tuning SVM Model	4 days	Mon 29/3/21	Thu 1/4/21

V. Gantt Chart for Project 2-Model Evaluation



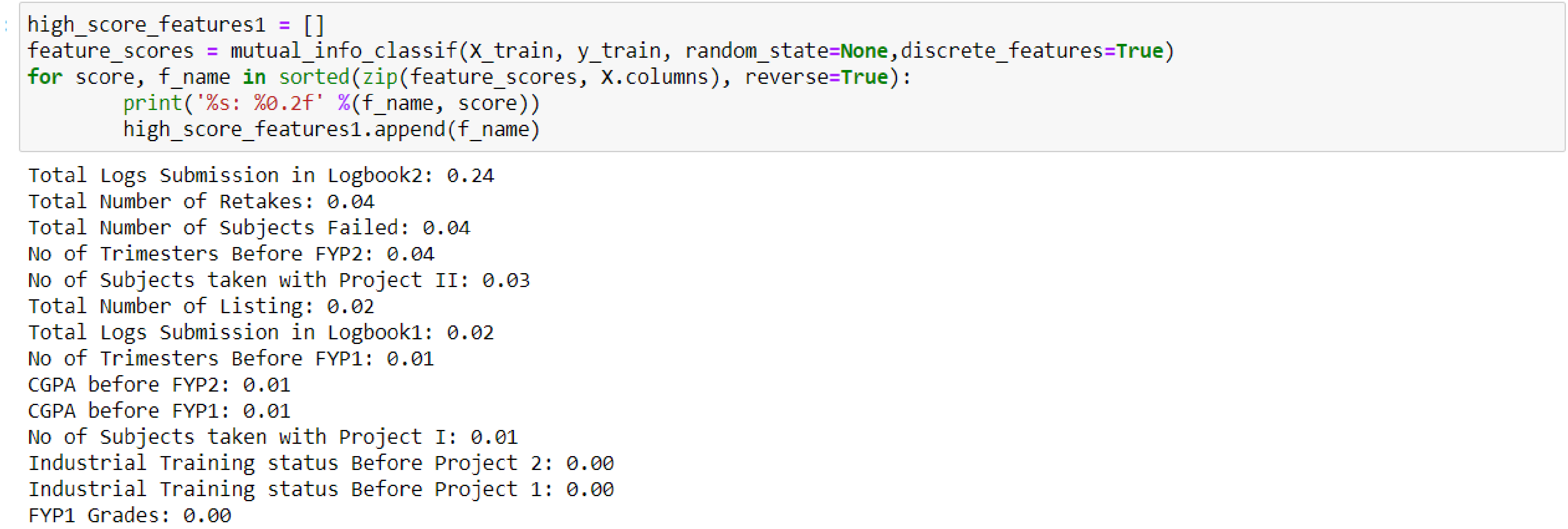
Task Name	Duration	Start	Finish
Project 2	**70 days**	**Mon 18/1/21**	**Fri 23/4/21**
Project Understanding	**5 days**	**Mon 18/1/21**	**Fri 22/1/21**
Data Collection/Understanding	**25 days**	**Mon 25/1/21**	**Fri 26/2/21**
Data Preparation	**5 days**	**Mon 1/3/21**	**Fri 5/3/21**
Modelling	**19 days**	**Mon 8/3/21**	**Thu 1/4/21**
Model Evaluation	**3 days**	**Fri 2/4/21**	**Tue 6/4/21**
Evaluation Using True Positive Rate of Fail Class	1 day	Fri 2/4/21	Fri 2/4/21
Evaluation Using True Positive Rate of Pass Class	1 day	Mon 5/4/21	Mon 5/4/21
Evaluation Using Balanced Accuracy	1 day	Tue 6/4/21	Tue 6/4/21

VI. Gantt Chart for Project 2-Model Evaluation



Task Name	Duration	Start	Finish
Project 2	**70 days**	**Mon 18/1/21**	**Fri 23/4/21**
Project Understanding	**5 days**	**Mon 18/1/21**	**Fri 22/1/21**
Data Collection/Understanding	**25 days**	**Mon 25/1/21**	**Fri 26/2/21**
Data Preparation	**5 days**	**Mon 1/3/21**	**Fri 5/3/21**
Modelling	**19 days**	**Mon 8/3/21**	**Thu 1/4/21**
Model Evaluation	**3 days**	**Fri 2/4/21**	**Tue 6/4/21**
Reporting	**13 days**	**Wed 7/4/21**	**Fri 23/4/21**
FYP Poster	1 day	Wed 7/4/21	Wed 7/4/21
Report Writing	7 days	Thu 8/4/21	Fri 16/4/21
Presentation Preparation	6 days	Sat 17/4/21	Fri 23/4/21

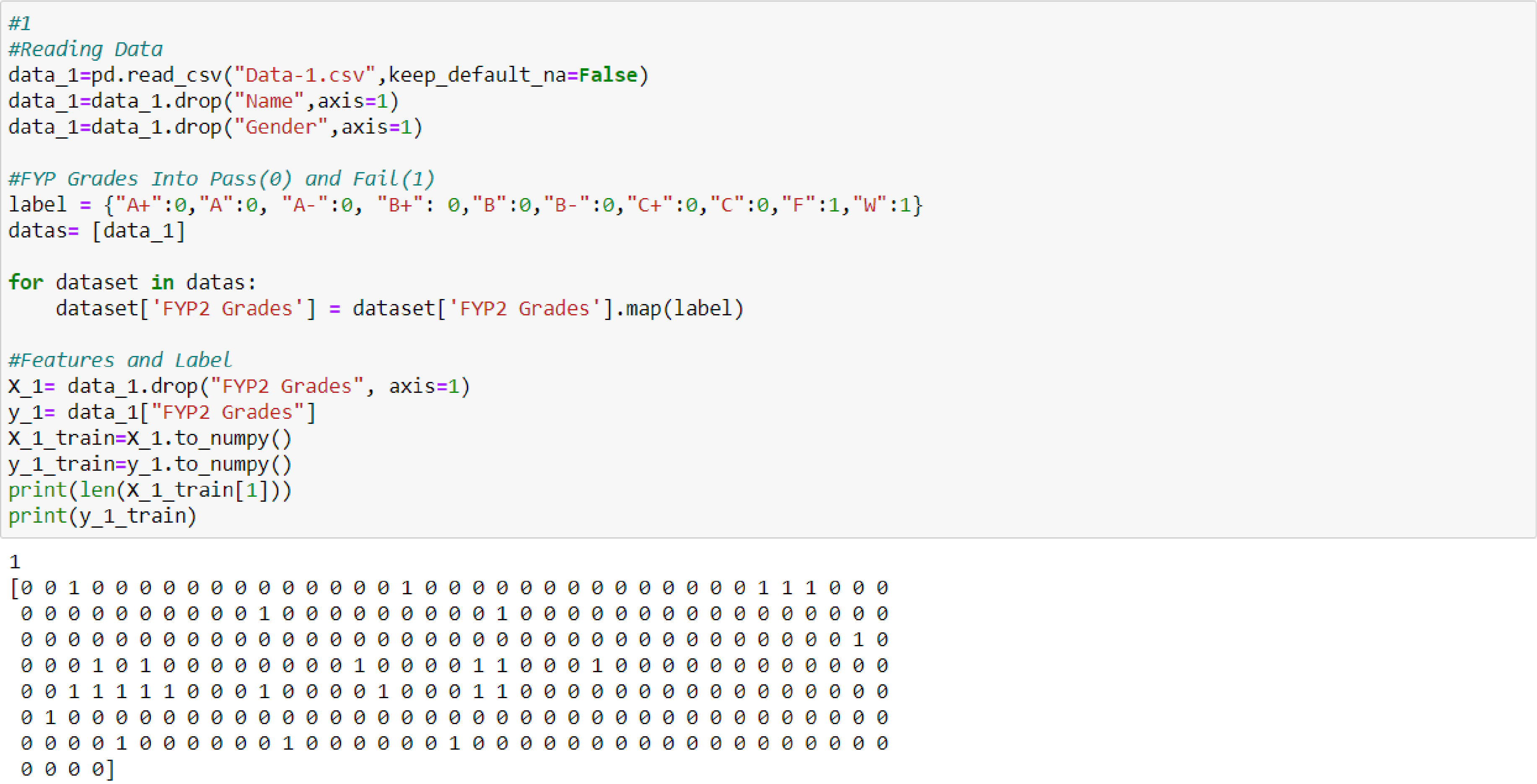
APPENDIX C: Sample Codes for Getting Mutual Information



```
high_score_features1 = []
feature_scores = mutual_info_classif(X_train, y_train, random_state=None,discrete_features=True)
for score, f_name in sorted(zip(feature_scores, X.columns), reverse=True):
        print('%s: %0.2f' %(f_name, score))
        high_score_features1.append(f_name)
```

```
Total Logs Submission in Logbook2: 0.24
Total Number of Retakes: 0.04
Total Number of Subjects Failed: 0.04
No of Trimesters Before FYP2: 0.04
No of Subjects taken with Project II: 0.03
Total Number of Listing: 0.02
Total Logs Submission in Logbook1: 0.02
No of Trimesters Before FYP1: 0.01
CGPA before FYP2: 0.01
CGPA before FYP1: 0.01
No of Subjects taken with Project I: 0.01
Industrial Training status Before Project 2: 0.00
Industrial Training status Before Project 1: 0.00
FYP1 Grades: 0.00
```

APPENDIX D: Sample Codes for Reading Data

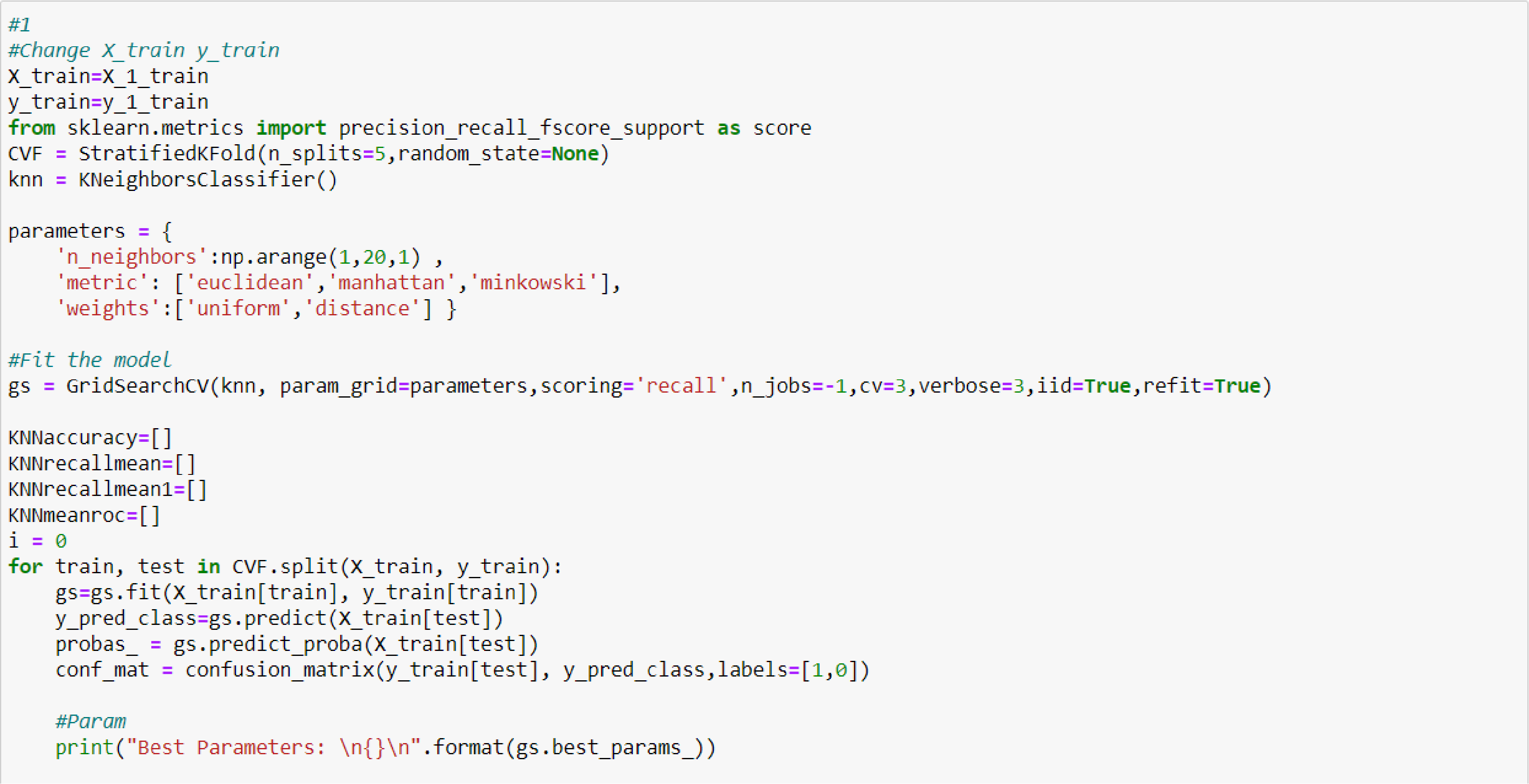


```
#1
#Reading Data
data_1=pd.read_csv("Data-1.csv",keep_default_na=False)
data_1=data_1.drop("Name",axis=1)
data_1=data_1.drop("Gender",axis=1)

#FYP Grades Into Pass(0) and Fail(1)
label = {"A+":0,"A":0, "A-":0, "B+": 0,"B":0,"B-":0,"C+":0,"C":0,"F":1,"W":1}
datas= [data_1]

for dataset in datas:
    dataset['FYP2 Grades'] = dataset['FYP2 Grades'].map(label)

#Features and Label
X_1= data_1.drop("FYP2 Grades", axis=1)
y_1= data_1["FYP2 Grades"]
X_1_train=X_1.to_numpy()
y_1_train=y_1.to_numpy()
print(len(X_1_train[1]))
print(y_1_train)
```

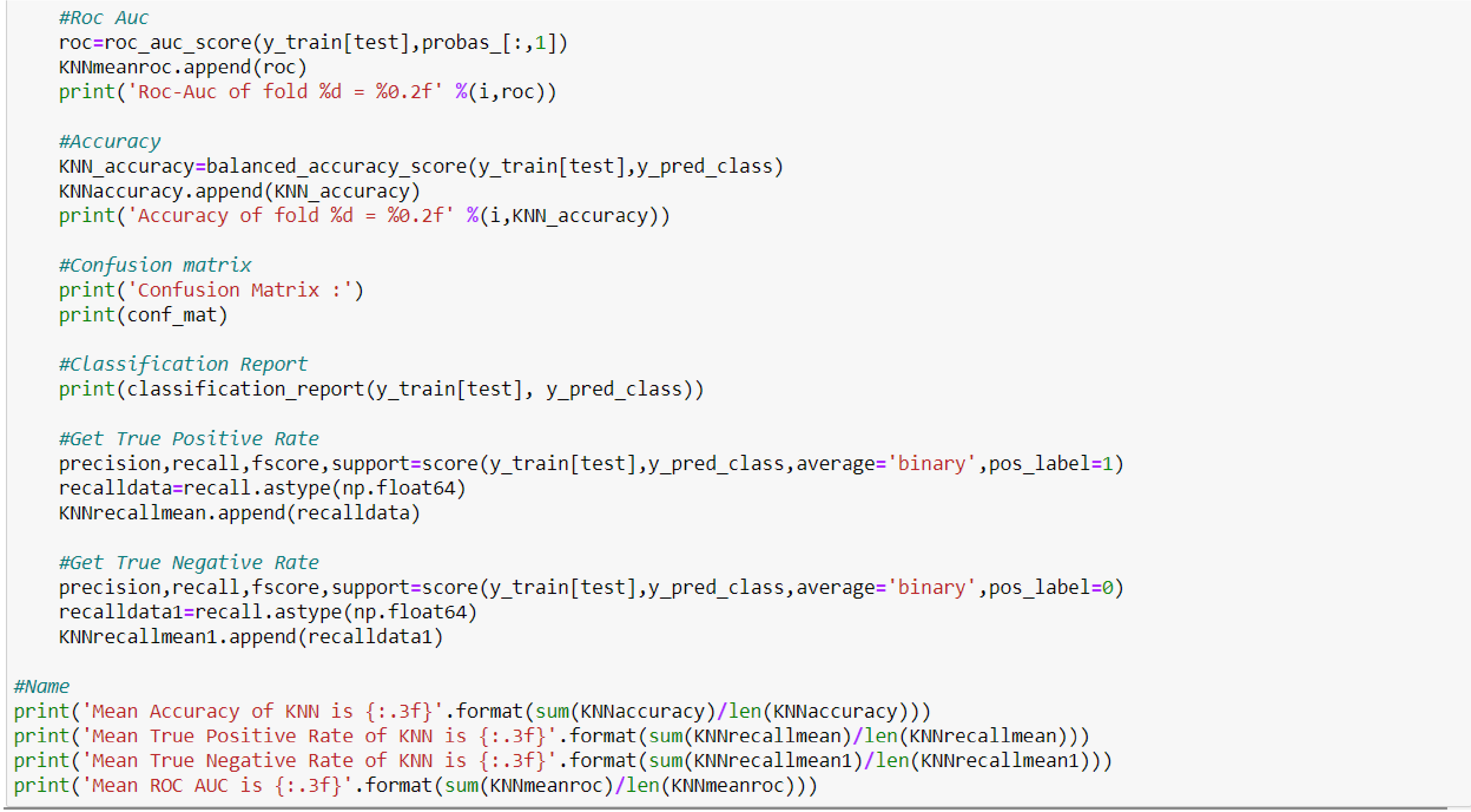
```
1
[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0
 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 1 1 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0]
```

APPENDIX E: Sample Codes for Modelling

KNN

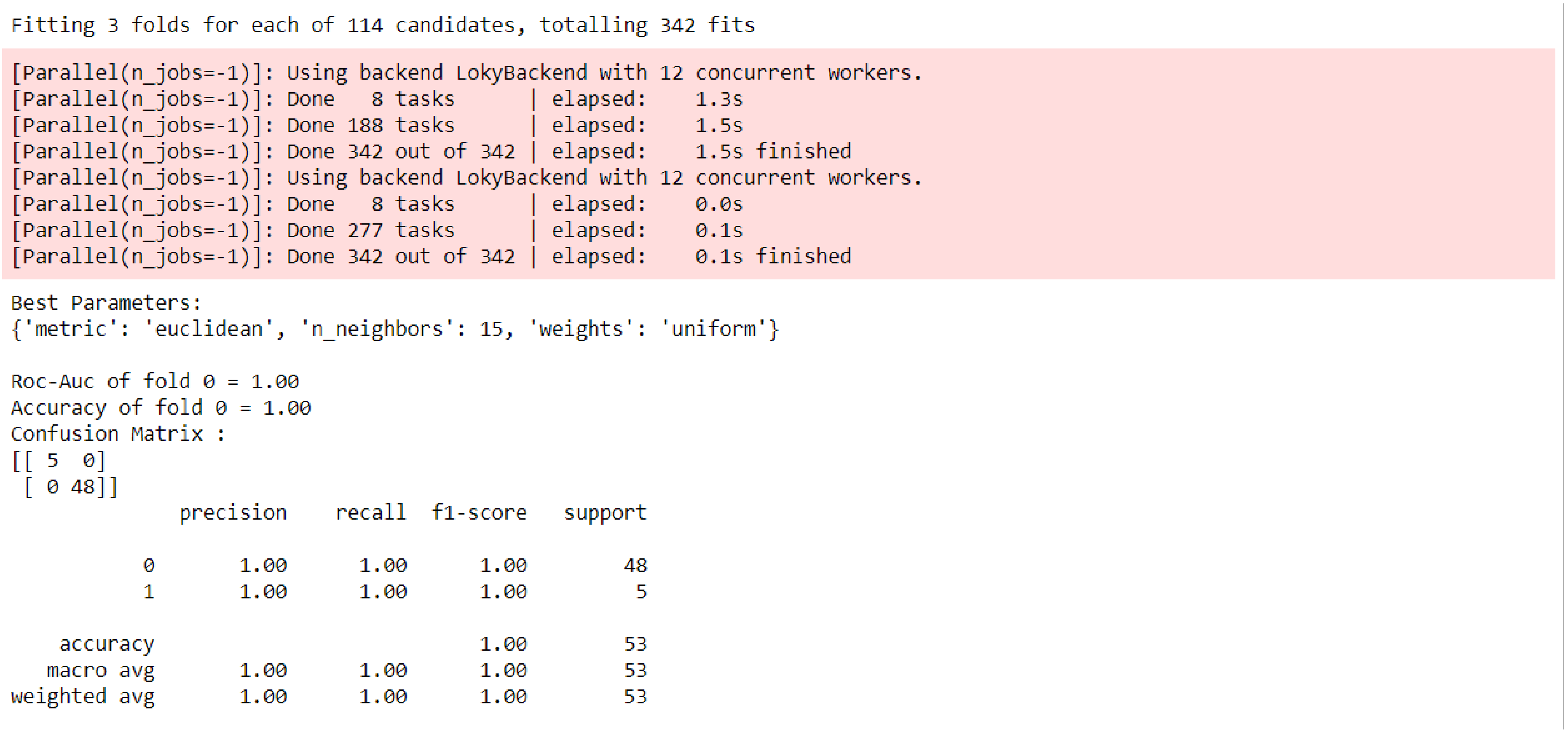


```
#1
#Change X_train y_train
X_train=X_1_train
y_train=y_1_train
from sklearn.metrics import precision_recall_fscore_support as score
CVF = StratifiedKFold(n_splits=5,random_state=None)
knn = KNeighborsClassifier()

parameters = {
    'n_neighbors':np.arange(1,20,1) ,
    'metric': ['euclidean','manhattan','minkowski'],
    'weights':['uniform','distance'] }

#Fit the model
gs = GridSearchCV(knn, param_grid=parameters,scoring='recall',n_jobs=-1,cv=3,verbose=3,iid=True,refit=True)

KNNaccuracy=[]
KNNrecallmean=[]
KNNrecallmean1=[]
KNNmeanroc=[]
i = 0
for train, test in CVF.split(X_train, y_train):
    gs=gs.fit(X_train[train], y_train[train])
    y_pred_class=gs.predict(X_train[test])
    probas_ = gs.predict_proba(X_train[test])
    conf_mat = confusion_matrix(y_train[test], y_pred_class,labels=[1,0])

    #Param
    print("Best Parameters: \n{}\n".format(gs.best_params_))
```



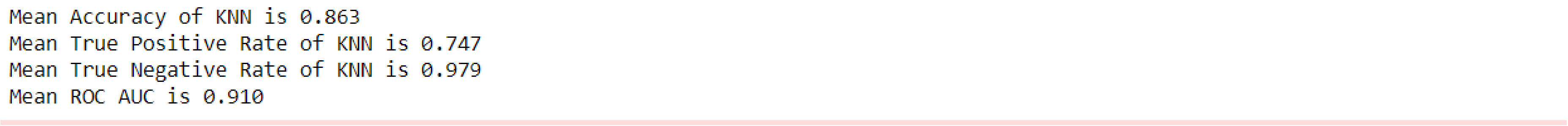
```
    #Roc Auc
    roc=roc_auc_score(y_train[test],probas_[:,1])
    KNNmeanroc.append(roc)
    print('Roc-Auc of fold %d = %0.2f' %(i,roc))

    #Accuracy
    KNN_accuracy=balanced_accuracy_score(y_train[test],y_pred_class)
    KNNaccuracy.append(KNN_accuracy)
    print('Accuracy of fold %d = %0.2f' %(i,KNN_accuracy))

    #Confusion matrix
    print('Confusion Matrix :')
    print(conf_mat)

    #Classification Report
    print(classification_report(y_train[test], y_pred_class))

    #Get True Positive Rate
    precision,recall,fscore,support=score(y_train[test],y_pred_class,average='binary',pos_label=1)
    recalldata=recall.astype(np.float64)
    KNNrecallmean.append(recalldata)

    #Get True Negative Rate
    precision,recall,fscore,support=score(y_train[test],y_pred_class,average='binary',pos_label=0)
    recalldata1=recall.astype(np.float64)
    KNNrecallmean1.append(recalldata1)

#Name
print('Mean Accuracy of KNN is {:.3f}'.format(sum(KNNaccuracy)/len(KNNaccuracy)))
print('Mean True Positive Rate of KNN is {:.3f}'.format(sum(KNNrecallmean)/len(KNNrecallmean)))
print('Mean True Negative Rate of KNN is {:.3f}'.format(sum(KNNrecallmean1)/len(KNNrecallmean1)))
print('Mean ROC AUC is {:.3f}'.format(sum(KNNmeanroc)/len(KNNmeanroc)))
```

APPENDIX F: Sample Output for Modelling Result(1-fold)



```
Fitting 3 folds for each of 114 candidates, totalling 342 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 12 concurrent workers.
[Parallel(n_jobs=-1)]: Done   8 tasks      | elapsed:    1.3s
[Parallel(n_jobs=-1)]: Done 188 tasks      | elapsed:    1.5s
[Parallel(n_jobs=-1)]: Done 342 out of 342 | elapsed:    1.5s finished
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 12 concurrent workers.
[Parallel(n_jobs=-1)]: Done   8 tasks      | elapsed:    0.0s
[Parallel(n_jobs=-1)]: Done 277 tasks      | elapsed:    0.1s
[Parallel(n_jobs=-1)]: Done 342 out of 342 | elapsed:    0.1s finished
Best Parameters:
{'metric': 'euclidean', 'n_neighbors': 15, 'weights': 'uniform'}

Roc-Auc of fold 0 = 1.00
Accuracy of fold 0 = 1.00
Confusion Matrix :
[[ 5  0]
 [ 0 48]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        48
           1       1.00      1.00      1.00         5

    accuracy                           1.00        53
   macro avg       1.00      1.00      1.00        53
weighted avg       1.00      1.00      1.00        53
```

APPENDIX G: Sample Output for Modelling Result(5-fold Average)



```
Mean Accuracy of KNN is 0.863
Mean True Positive Rate of KNN is 0.747
Mean True Negative Rate of KNN is 0.979
Mean ROC AUC is 0.910
```