



# UNIVERSITI MALAYA

## WIE3007 DATA MINING AND WAREHOUSING SEMESTER 1 2023/2024

Group Members:

Name	Matric ID
Au Wan Ying	U2005373/1
Chuah Ann Joe	U2005355/1
Lee Xiao Yu	U2005405/1
Ruo Jun Wang	S2011618
Tiew Ker Xin	U2005253/1

# Table of Contents

<b>Table of Contents</b> .....	<b>2</b>
<b>1.0 Business Objectives</b> .....	<b>3</b>
<b>2.0 Introduction</b> .....	<b>4</b>
<b>3.0 Sample</b> .....	<b>6</b>
<b>4.0 Featuretools And Star Schema</b> .....	<b>10</b>
4.1 Featuretools.....	10
4.2 Star Schema.....	14
<b>5.0 Explore</b> .....	<b>15</b>
5.1 Extract Data From CSV File Using Talend Open Studio.....	15
5.2 Exploratory Data Analysis.....	15
5.2.1 Impact Of Debt On Student Target.....	16
5.2.2 Correlation of Age At Enrollment With First And Second Semester Grade.....	17
5.2.3 Academic Outcomes By Course.....	18
<b>6.0 Modify</b> .....	<b>20</b>
6.1 Data Preprocessing.....	20
<b>7.0 Techniques And Algorithms</b> .....	<b>27</b>
7.1 Association Rule.....	27
7.2 Sequence Analysis.....	30
7.3 DBSCAN (Density-Based Spatial Clustering of Applications with Noise).....	34
<b>8.0 Model</b> .....	<b>37</b>
8.1 Decision Tree.....	37
8.2 Regression.....	41
8.3 Gradient Boosting.....	43
8.4 Random Forest.....	44
<b>9.0 Assess</b> .....	<b>47</b>
<b>10.0 Conclusion</b> .....	<b>52</b>
<b>11.0 References</b> .....	<b>54</b>

## **1.0 Business Objectives**

1. To investigate the factors that contribute to the academic dropouts.
2. To optimise the curriculum performance based on the identified factors.
3. To foster a collaborative learning environment based on the factors that contribute to the academic success

## 2.0 Introduction

In the realm of higher education, the growing concern of student dropout and academic failure has prompted the exploration of innovative solutions. This paper delves into a project focused on addressing the challenges of reducing dropout and failure rates among students in higher education institutions. The primary focus is on early identification of at-risk students, allowing timely implementation of targeted strategies to support their academic journey.

The dataset utilized for this endeavour originates from a comprehensive project aimed at deciphering patterns in student behaviour and performance. Compiled over a period that spans the initial stages of academic enrollment, the dataset captures a wealth of information encompassing academic pathways, demographics, and socio-economic factors. It provides a holistic view of students' backgrounds and circumstances at the time of their entry into higher education institutions.

The problem at hand is formulated as a three-category classification task, involving the prediction of student outcomes classified into dropout, enrolment, and postgraduate categories. The dataset is derived from diverse higher education institutions, encompassing a spectrum of undergraduate degrees, including but not limited to agriculture, design education, nursing, journalism, and management.

With a robust dataset containing 4424 instances and 36 features, this project aims to unravel predictive insights. By leveraging this wealth of information, the goal is to discern early indicators of potential dropout or academic success. This proactive approach allows educational institutions to intervene promptly and implement tailored strategies to mitigate the risk of student attrition.

As the education landscape continues to evolve, understanding and addressing the factors contributing to student dropout become crucial for fostering a supportive environment. This project contributes to the ongoing discourse on enhancing educational outcomes by investigating the nuances of student pathways and identifying key factors that influence academic success.

**Below is the link of dataset being used in this project:**

<http://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success>

### 3.0 Sample

Google Colab Link For Sampling:

[https://colab.research.google.com/drive/1n9JbekNn47f\\_ndF1n5FPF6cHGslYEHZ8?usp=sharing](https://colab.research.google.com/drive/1n9JbekNn47f_ndF1n5FPF6cHGslYEHZ8?usp=sharing)

In this sampling part, there are three sampling methods applied to read the data directly using the pandas library, which are random sampling according to proportion, grouping according to different categories of the specified columns, and then randomly sampling the data within each group. As well as random sampling with weights in accordance with the weights. described one at a time below. The code reads the data as shown in Diagram 3.1:

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Previous qualification (grade)	Nacionality	Mother's qualification	Father's qualification
0	1	17	5	171	1	1	122.0	1	19	
1	1	15	1	9254	1	1	160.0	1	1	
2	1	1	5	9070	1	1	122.0	1	37	
3	1	17	2	9773	1	1	122.0	1	38	
4	2	39	1	8014	0	1	100.0	1	37	
...	...	...	...	...	...	...	...	...	...	...
4419	1	1	6	9773	1	1	125.0	1	1	
4420	1	1	2	9773	1	1	120.0	105	1	
4421	1	1	1	9500	1	1	154.0	1	37	
4422	1	1	1	9147	1	1	180.0	1	37	
4423	1	10	1	9773	1	1	152.0	22	38	

4424 rows × 11 columns

Diagram 3.1 : Raw Data

To use random sampling, we need to pre-select in advance the proportion of the sample, which is a decimal number between 0 and 1 that indicates the proportion of the original data you want to sample. For example, if the proportion is 0.2, then 20% of the original data will be sampled. It is important to note here that random state This is an optional parameter to set the random number seed to ensure that you get the same random sampling result every time you run the code. If you want

the results to be reproducible, you can specify a fixed random number seed. The sampling results are shown in Diagram 3.2:

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Previous qualification (grade)	Nacionality	Mother's qualification	Fathe qualificati
1255	4	39	1	9130	1	1	133.1	1	3	
3458	1	17	1	9238	1	1	125.0	1	4	
3390	1	17	1	9853	1	1	133.0	1	38	
1497	1	17	2	9670	1	1	110.0	1	1	
1536	1	39	1	9500	1	1	130.0	1	37	
...	...	...	...	...	...	...	...	...	...	...
4235	1	1	3	9238	1	1	133.1	1	1	
979	1	18	1	9003	1	1	133.1	1	1	
283	1	1	1	9070	1	1	126.0	1	19	
1298	1	1	1	9991	0	1	145.0	1	2	
898	1	1	1	9254	1	1	112.0	1	38	

1327 rows × 37 columns

**Diagram 3.2 : Random Sampling Results**

Grouping and sampling by different categories of a specified column uses the groupby and apply methods in Pandas to group the data in a DataFrame by different categories of a given column and then randomly sample the data within each group. It should be emphasised that this sampling method ensures a balanced number of samples from each category because the number of samples from each group is set. The sampling results are shown in Diagram 3.3:

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Previous qualification (grade)	Nationality	Mother's qualification	Father's qualification	...	Curricular units 2nd sem (credited)	Curricular units (enrolled)	
Marital status	1	3290	1	1	9238	1	1	125.0	1	1	1	...	0	
		3416	1	1	9773	1	1	143.0	1	3	3	...	0	
		2522	1	17	6	9500	1	1	131.0	1	1	19	...	0
		4135	1	1	1	8014	0	1	115.0	1	38	38	...	0
		981	2	39	1	9991	0	1	100.0	1	19	19	...	0
	2	2006	2	39	1	8014	0	1	146.0	1	37	37	...	0
		4034	2	39	1	8014	0	19	133.1	1	34	34	...	0
		1739	2	39	1	9070	1	19	133.1	1	37	37	...	0
		688	3	39	1	9991	0	1	120.0	1	37	1	...	0
	3	166	3	39	1	9003	1	1	170.0	1	1	37	...	0
		1428	3	17	1	9238	1	1	138.0	1	37	37	...	0
		428	3	1	2	9085	1	1	135.0	1	19	1	...	0
		409	4	39	1	9085	1	1	133.1	1	34	34	...	0
	4	939	4	39	1	8014	0	19	133.1	1	37	37	...	0
		2999	4	39	1	8014	0	1	138.0	1	19	19	...	0
		2794	4	39	1	9991	0	1	130.0	1	37	37	...	0
		4374	5	7	1	9500	1	40	150.0	1	37	37	...	1
	5	3943	5	39	1	9147	1	1	180.0	1	1	1	...	0
		3721	5	7	1	9070	1	3	140.0	1	37	37	...	0
		2919	5	39	1	9003	1	19	133.1	1	19	1	...	0
		2914	6	39	2	9147	1	1	133.1	1	37	37	...	0
	6	2915	6	39	1	8014	0	1	133.1	1	37	37	...	0
		1180	6	39	1	9500	1	1	133.1	1	37	37	...	0
		1572	6	39	1	9991	0	12	130.0	1	37	37	...	0

24 rows x 37 columns

24 rows x 37 columns

**Diagram 3.3 : Sampling results of specified categories**

The last method is random sampling with weights in accordance with the weights, where it is necessary to define a list of weights in advance, which indicates the probability of each sample being drawn. The higher the weight, the higher the probability of being drawn, is a list of the same size as the dataset, which is used to specify the weight of each sample. The role of the entire code is to randomly sample from the data with weights, the number of samples is controlled by num, and the random state parameter ensures that the sampling results are reproducible. The sampling results are shown in Diagram 3.4 :



	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Previous qualification (grade)	Nacionality	Mother's qualification	Father's qualification	...	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	(e)
1681	1	1	1	9773	1	1	117.0	1	1	19	...	0	6	
4212	1	17	2	9773	1	1	125.0	1	1	1	...	0	6	
3261	1	1	3	9147	1	1	115.0	1	1	38	...	0	5	
2667	2	39	1	9991	0	19	133.1	1	19	19	...	0	5	
706	2	39	1	8014	0	1	133.1	1	19	1	...	3	7	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
872	1	42	1	9991	0	1	150.0	1	19	19	...	6	11	
1516	1	7	1	9500	1	5	140.0	1	19	1	...	0	8	
3799	2	39	1	9085	1	12	133.1	1	1	39	...	0	6	
964	1	1	1	9773	1	1	121.0	1	1	3	...	0	6	
321	1	42	1	9500	1	1	100.0	1	37	37	...	0	8	

2212 rows × 37 columns

**Diagram 3.4 : Weight sampling results**

## 4.0 Featuretools And Star Schema

### 4.1 Featuretools

Google Colab Link For Featuretools:

<https://colab.research.google.com/drive/1SfNII4Mzsdhf-3HGgg1WUGhbmFkBEodn?usp=sharing>

```
import featuretools as ft
import pandas as pd

# Specify the path to your CSV file
csv_file_path =
'C:\\Users\\kerxi\\OneDrive\\Documents\\Desktop\\WIE3007 DATA MINING
AND WAREHOUSING\\Group Assignment\\sampled_data.csv'

# Use pandas to read the CSV file
data = pd.read_csv(csv_file_path)

# Define entities
student_entity = (data[['MaritalStatus', 'Nationality', 'Gender',
'ScholarshipHolder', 'International',

'DaytimeEveningAttendance', 'Debtor', 'PreviousQualification',
'PreviousQualificationGrade',
'MotherQualification', 'FatherQualification',
'MotherOccupation', 'FatherOccupation',
'Displaced',
'Student_ID']], 'Student_ID')

enrollment_entity = (data[['AdmissionGrade', 'EducationalSpecialNeeds',

'TuitionFeesUpToDate', 'AgeAtEnrollment', 'Student_ID',
'Enrollment_ID']], 'Enrollment_ID')

performance_entity = (data[['CurricularUnits1stSemCredited',
'CurricularUnits1stSemEnrolled',
'CurricularUnits1stSemEvaluation',
```

```

        'CurricularUnits1stSemApproved', 'CurricularUnits1stSemGrade',
        'CurricularUnits1stSemNoEvaluations',
'CurricularUnits2ndSemCredited',
        'CurricularUnits2ndSemEnrolled',
'CurricularUnits2ndSemEvaluations',
        'CurricularUnits2ndSemApproved', 'CurricularUnits2ndSemGrade',
        'CurricularUnits2ndSemWithoutEvaluations','UnemploymentRate',
                'InflationRate', 'GDP', 'Target',
'Performance_ID','Student_ID']], 'Performance_ID')

```

```

course_entity = (data[[ 'Course','Course_ID','Student_ID']],
'Course_ID')

```

```

application_entity = (data[[ 'ApplicationMode', 'ApplicationOrder',
'Application_ID','Student_ID']], 'Application_ID')

```

```

# Create an EntitySet

```

```

es = ft.EntitySet(id="student_data")

```

```

# Add dataframes to the EntitySet

```

```

es = es.add_dataframe(dataframe_name="student",
dataframe=student_entity[0], index='Student_ID')

```

```

es = es.add_dataframe(dataframe_name="enrollment",
dataframe=enrollment_entity[0], index='Enrollment_ID')

```

```

es = es.add_dataframe(dataframe_name="student_performance",
dataframe=performance_entity[0], index='Performance_ID')

```

```

es = es.add_dataframe(dataframe_name="course",
dataframe=course_entity[0], index='Course_ID')

```

```

es = es.add_dataframe(dataframe_name="application",
dataframe=application_entity[0], index='Application_ID')

```

```

#Establish relationship

```

```

relationship1 = es.add_relationship(
    parent_dataframe_name='student',

```

```

        parent_column_name='Student_ID',
        child_dataframe_name='student_performance',
        child_column_name='Student_ID'
    )

relationship2 = es.add_relationship(
    parent_dataframe_name='student',
    parent_column_name='Student_ID',
    child_dataframe_name='enrollment',
    child_column_name='Student_ID'
)

relationship3 = es.add_relationship(
    parent_dataframe_name='student',
    parent_column_name='Student_ID',
    child_dataframe_name='course',
    child_column_name='Student_ID'
)

relationship4 = es.add_relationship(
    parent_dataframe_name='student',
    parent_column_name='Student_ID',
    child_dataframe_name='application',
    child_column_name='Student_ID'
)

# Print the EntitySet
print(es)

# SET PANDAS DISPLAY OPTIONS TO SHOW ALL COLUMNS AND ROWS
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

# DEEP FEATURE SYNTHESIS
feature_matrix, feature_defs = ft.dfs(
    entityset=es,
    target_dataframe_name="student_performance",
    verbose=True,
    max_depth=2
)

```

```
feature_matrix, feature_defs = ft.dfs(  
    entityset=es,  
    target_dataframe_name="enrollment",  
    verbose=True,  
    max_depth=2  
)  
  
# PRINT THE GENERATED FEATURE MATRIX  
print(feature_matrix)  
print(feature_defs)
```

## 4.2 Star Schema

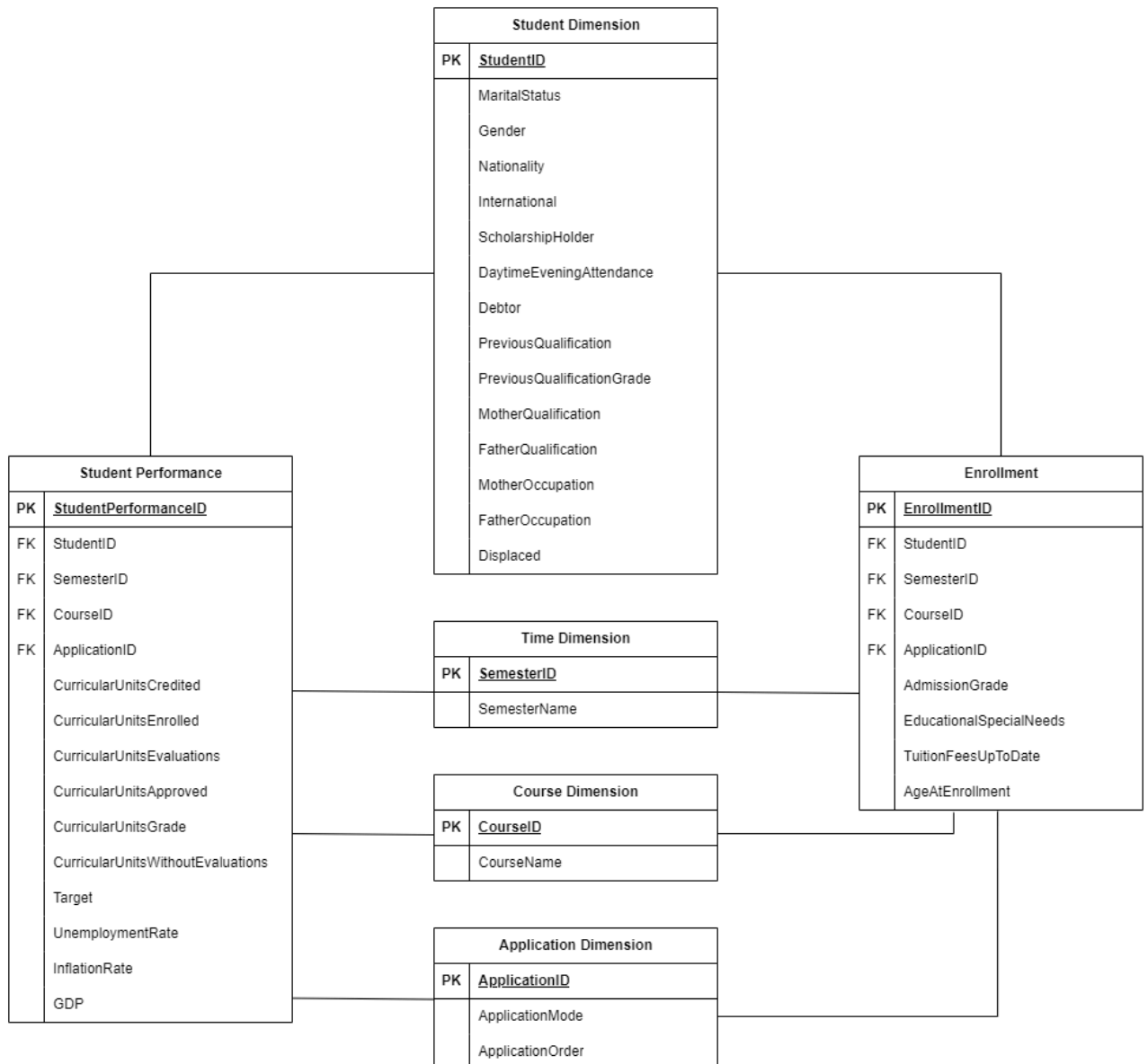
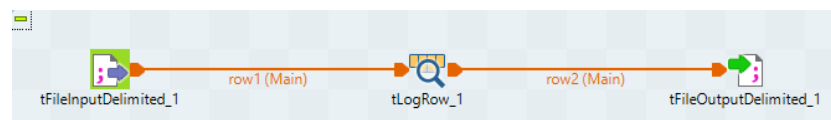


Diagram 4.1 Star Schema

## 5.0 Explore

### 5.1 Extract Data From CSV File Using Talend Open Studio

Before exploratory data analysis has been carried out, Talend Open Studio is used to extract necessary information. tFileInputDelimited component is dragged into the workspace to read the CSV file. Configuration is carried out and the required columns in the CSV file are defined by inserting column names and specifying the data types. Some columns such as Student\_ID, Enrollment\_ID, Performance\_ID, Course\_ID, and Application\_ID which are initially being used in Featuretools but unnecessary for the proceeding steps are removed. tLogRow component is used to view the extracted data from CSV. After that, tFileOutputDelimited is used to extract the new CSV file that contains only the columns which are needed for the SEMMA process.



**Diagram 5.1 Component Used For Data Extraction**

### 5.2 Exploratory Data Analysis

Exploratory data analysis (EDA) usually involves the use of data visualisation techniques to examine, analyse, and summarise key features of datasets. It assists in providing the best method to manipulate data sources in obtaining the desired answers. Therefore, it makes the task of finding patterns, identifying anomalies, testing hypotheses, and verifying assumptions becomes easier. Before proceeding to data analysis, EDA helps to determine whether the selected statistical techniques are appropriate.

### 5.2.1 Impact Of Debt On Student Target

The diagram 5.2 below shows the bar chart plotting the impact of debt on student targets. For x-axis, value 0 represents that the students are non-debtors while value 1 indicates that the students are debtors. There are a total of 1165 students who are non-debtors while 161 students are debtors. Among the non-debtor, the graduation rate is higher than the dropout rate whereas the dropout rate for debtors is higher than that of the graduation rate. Therefore, we can conclude that the status of debt of a student will influence the academic dropout.

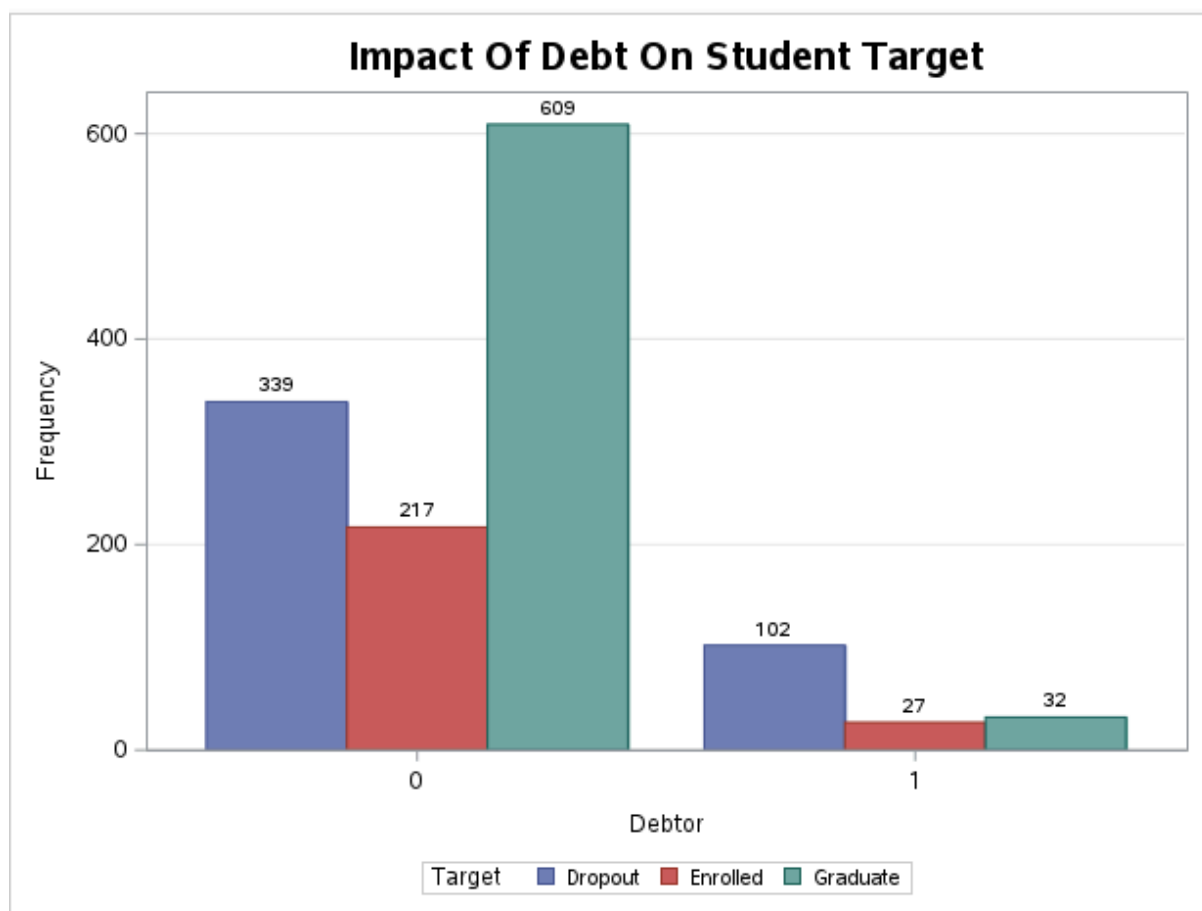


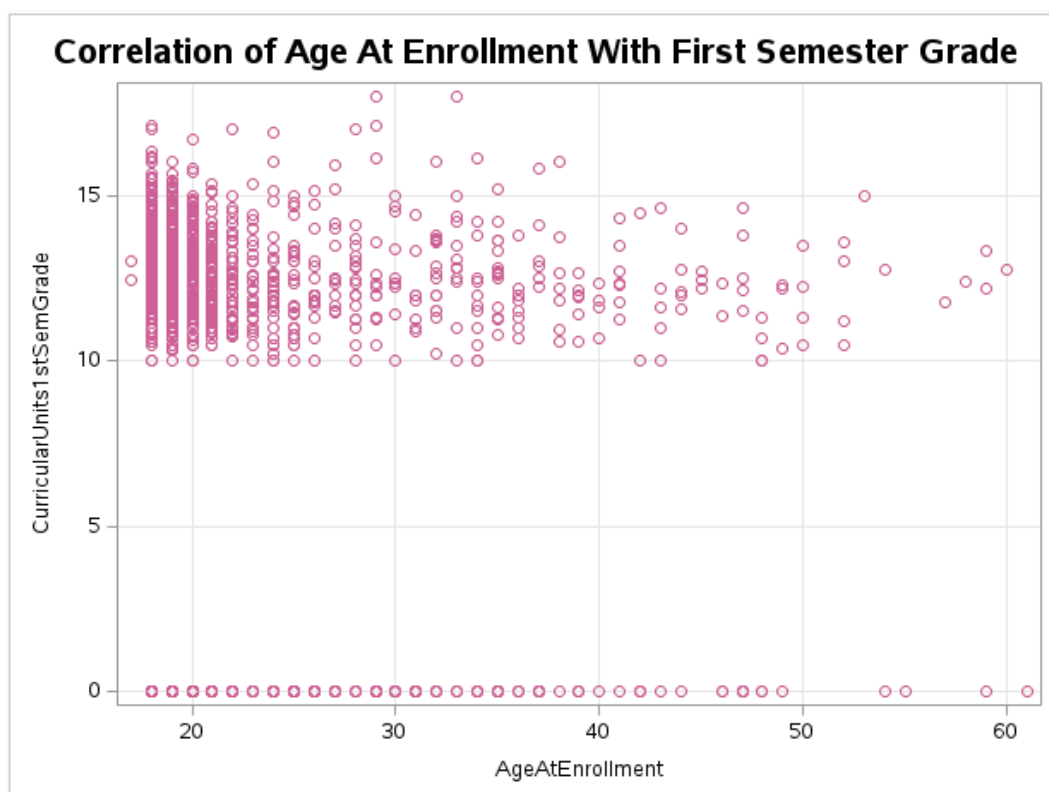
Diagram 5.2 Impact Of Debt On Student Target



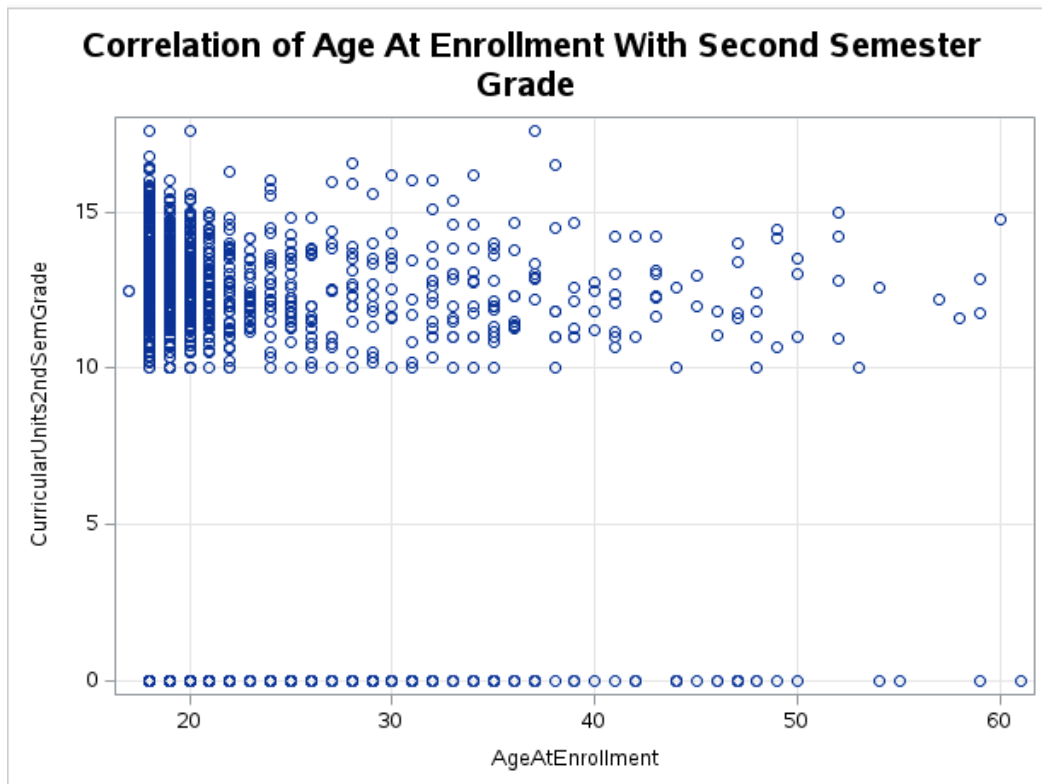
## 5.2.2 Correlation of Age At Enrollment With First And Second Semester Grade

The diagram 5.3 below shows the scatter plot of correlation of age at enrollment with first semester grade while diagram 5.4 shows the scatter plot of correlation of age at enrollment with second semester grade. From both diagrams, we can see that the age range of the majority of the students is 18 - 30 while the number of students decreased by age. Most of the students with age of enrollment < 30 achieve a higher grade than students with other ages. This might be due to younger students having more time to study due to less external commitments such as work and family responsibilities.

By identifying age at enrollment as a factor that will affect the curriculum performance, educators can prepare different additional resources, study plans, or teaching methods to enhance the overall performance.



**Diagram 5.3 Correlation of Age At Enrollment With First Semester Grade**



**Diagram 5.4 Correlation of Age At Enrollment With Second Semester Grade**

### 5.2.3 Academic Outcomes By Course

Diagram 5.5 below shows the heat map plotting the academic outcomes by courses. Different colours represent the number of students who take different courses with different targets. This graph allows the investigation of student targets who take different courses. Most of the students who take courses with course code 9500 (Nursing) have the highest rate of graduation, followed by course code 33 (Biofuel Production Technologies) and 9238 (Social Service). Although the nursing course (course code 9500) possesses the highest graduation rate, it also has the highest dropout rate among other courses. However, courses 9500 (Nursing), 9147 (Management) and 9119 (Informatics Engineering) jotted the highest enrollment rate. The knowledge of academic outcomes by courses allows the school administration to make amendments and know which courses should put in effort in order to make improvement. As a result, a more collaborative learning environment is formed.

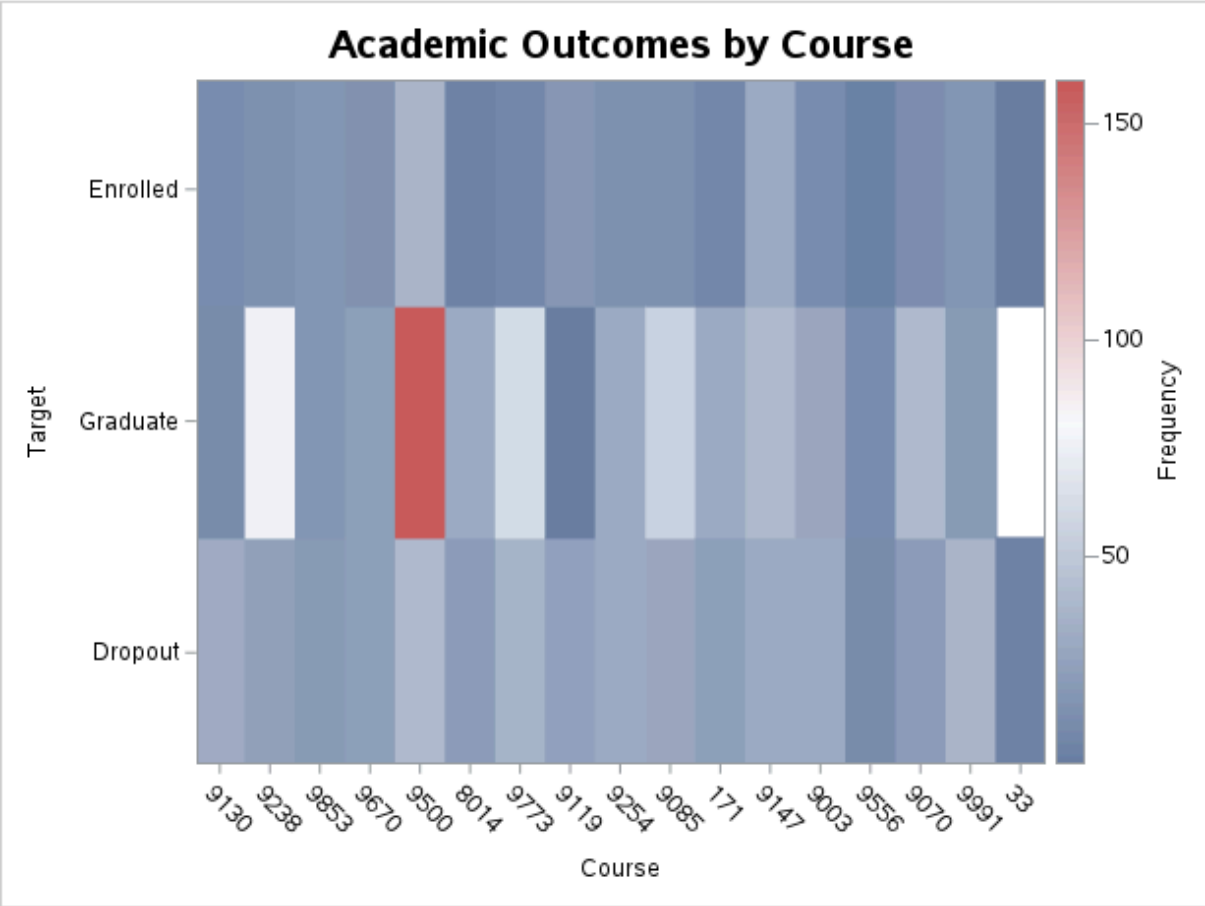
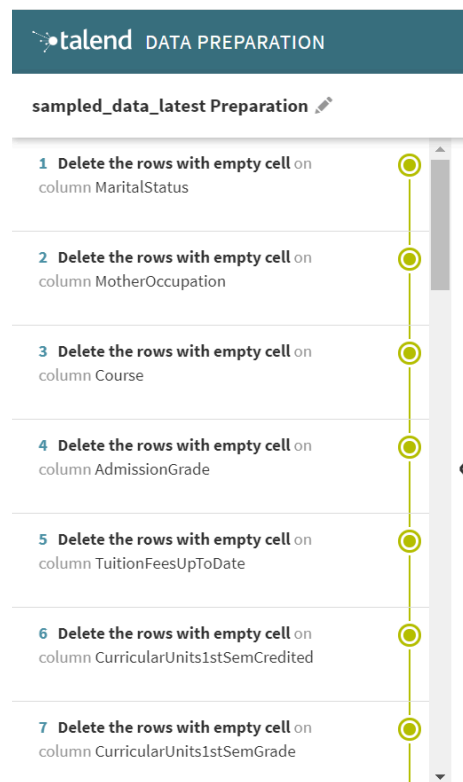


Diagram 5.5 Academic Outcomes By Courses

## 6.0 Modify

### 6.1 Data Preprocessing

Data preprocessing is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate or incomplete data within a dataset. Data preprocessing helps to ensure the quality of the data, correct errors, handle missing values as to prepare the dataset for further exploration. In this project, we choose to integrate Talend Data Preparation as the tool to perform the data preprocessing task on the chosen dataset.



**Diagram 6.1 Handling missing values**

The dataset is loaded into Talend Data Preparation to perform data preprocessing. The first task involves handling missing values within the dataset. In this process, rows containing missing values are directly removed. This decision is based on the assumption that the missing data occur at random. With the removal of missing values, it helps us to ensure the integrity and quality of the dataset for the

subsequent analysis. This approach is chosen to avoid biases that might result from imputing missing values, particularly in cases when the missingness is assumed to be random. Additionally, the outliers have not been explicitly detected in the dataset.

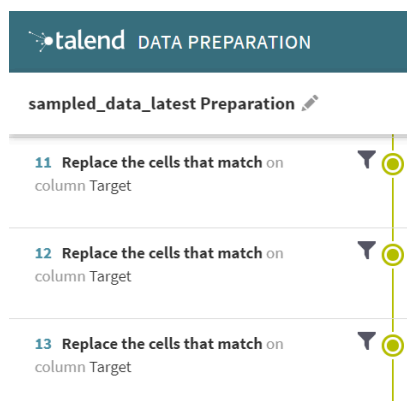


Diagram 6.2 Encode categorical variables

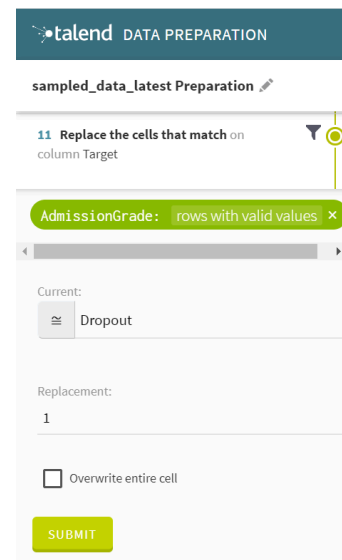


Diagram 6.3 “1” is assigned for “Dropout”

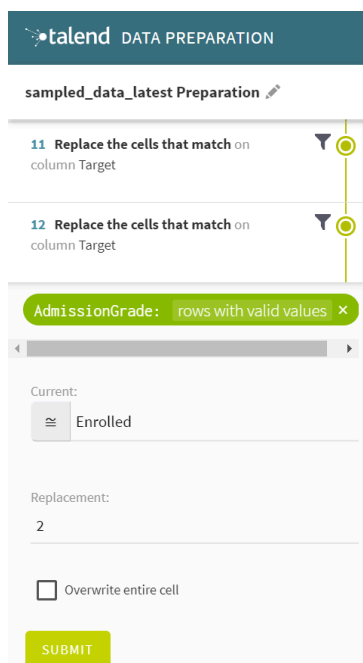


Diagram 6.4 “2” for “Enrolled”

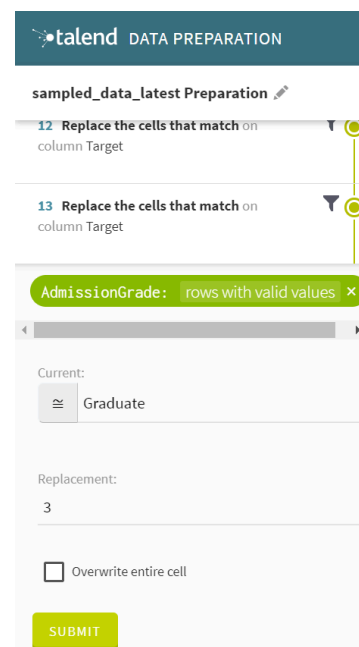
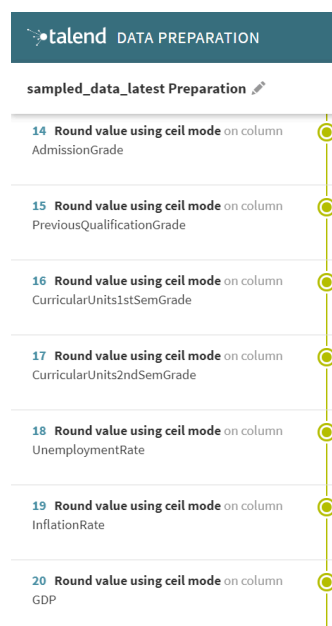


Diagram 6.5 “3” for “Graduate”

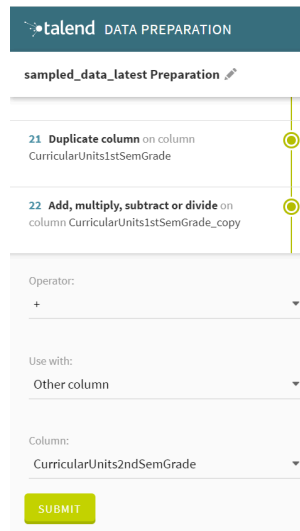
After the missing values are addressed, the next step involves encoding categorical variables in the dataset. Categorical variables need to be encoded into

numerical representations to facilitate model training, enabling effective interpretation and learning from these variables. In our chosen dataset, the “Target” attribute is the categorical variable. This attribute consists of three statuses: “Dropout”, “Enrolled” and “Graduate”. These categories are encoded using numerical representations. Specifically, “1” is assigned to represent the presence of “Dropout”, “2” for “Enrolled” and “3” for Graduate.

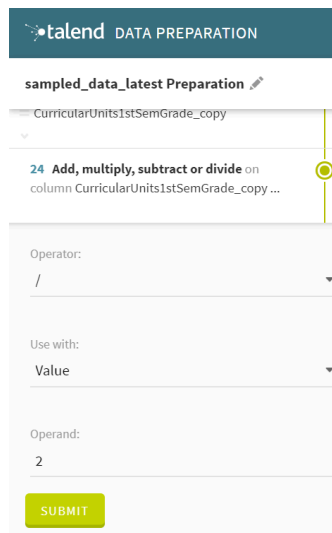


**Diagram 6.6 Standardization data by rounding up decimal places**

After encoding the categorical variable, the dataset undergoes standardization by rounding up the decimal places to two digits for the data. This step helps to ensure the uniformity in the representation of numerical values, as some data points may originally vary in precision with different numbers of decimal places. With standardization of data, it helps to mitigate the variations in the scale of numerical features that could potentially affect the performance of the model.



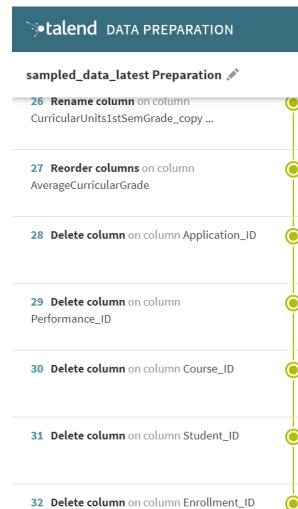
**Diagram 6.7 Summing the grades of CurricularUnits1stSemGrade and CurricularUnits2ndSemGrade**



**Diagram 6.8 Result obtained from Diagram 6.7 is divided by 2**

After standardizing the data, the next step is to perform feature engineering. Feature engineering is the process of creating new features by combining or transforming the existing ones to enhance the performance of the models. Feature engineering is performed on the chosen dataset by creating a new feature named "AverageCurricularGrade". This feature is derived by summing the grades of CurricularUnits1stSemGrade with CurricularUnits2ndSemGrade and dividing the result by 2. Incorporating the "AverageCurricularGrade" through feature engineering

enriches the dataset, providing the model with a refined input that captures key information from the original features.



**Diagram 6.9 Remove unused identifiers**

In addition, identifiers such as Application\_ID, Enrollment\_ID, Performance\_ID and Student\_ID are removed from the dataset. These identifiers generally do not contribute any informative value to the predictive tasks. By eliminating them, the dataset is streamlined and the analysis is concentrated on the relevant features which are essential for predictive modelling.

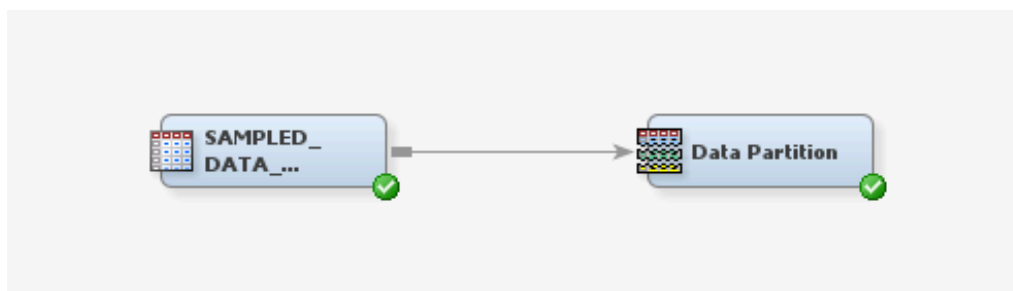
Furthermore, the original variable names in the dataset did not align with the requirements of SAS Enterprise Miner. As a result, some attributes were renamed to meet the compatibility criteria of the modelling tool. With these adjustments, the dataset is now optimally prepared for modelling purposes and ready to be exported for analysis.



```
Program 1 x
CODE LOG RESULTS
1 /* Generated Code (IMPORT) */
2 /* Source File: iris.csv */
3 /* Source Path: /home/u63663176/3010 Tutorial 11/DataSources */
4 /* Code generated on: 1/10/24, 9:31 AM */
5
6 LIBNAME mylib '/home/u63663171/Project/DataSources';
7
8 DATA mylib.Sampled_Data_LatestVersion;
9 SET WORK.IMPORT;
10 RUN;
```

**Diagram 6.10 Code to convert the uploaded dataset to sas7bdat format**

The preprocessed dataset is then uploaded into the SAS studio. The dataset is initially in xlsx format, which isn't directly supported by SAS Enterprise Miner for library creation. To address this, a coding solution is implemented to convert the uploaded dataset to the sas7bdat format. The specific code for this conversion is detailed in Diagram 6.10, ensuring the compatibility with the SAS Enterprise Miner and facilitating seamless retrieval for further analysis.



**Diagram 6.11 Data Partition node is used**

Then, a new data source is created by using the preprocessed dataset and it acts as the input in the model diagram. The dataset undergoes further preprocessing within the first node, where the "Target" is set as the Target while the other variables are set as the Input and the "ApplicationOrder" is set as Rejected. The dataset is then connected to the Data Partition node, responsible for allocating the data. 70% of the data is assigned into the training set for preliminary model fitting while dedicating 30% to the validation set for assessing the appropriateness of the model chosen.

#### Partition Summary

Type	Data Set	Number of Observations
DATA	EMWS2.Ids_DATA	1317
TRAIN	EMWS2.Part_TRAIN	922
VALIDATE	EMWS2.Part_VALIDATE	395

**Diagram 6.12 Data Partition Summary**

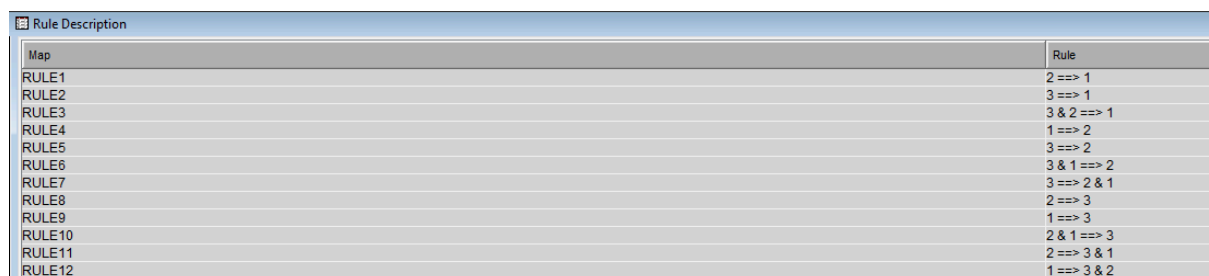
Diagram 6.12 shows the data partitioned for 70% train and 30% validate. The partition scheme is a crucial step in the modelling process, ensuring a comprehensive evaluation of the model's performance.

## 7.0 Techniques And Algorithms

### 7.1 Association Rule

Association Rule is a data mining technique that helps to illustrate the relationship between various variables in a dataset. It is frequently employed in market basket analysis to find trends in consumer purchasing behaviour. The association rule in this project is being carried out by using courses as ID and the target of student (dropout, enrolled and graduate) as the target to look into any particular courses or combinations of courses that may be linked to increased graduation rates, dropout rates, or other student outcomes. Moreover, association rule mining also helps to estimate the probability of various student outcomes according to the courses they have taken.

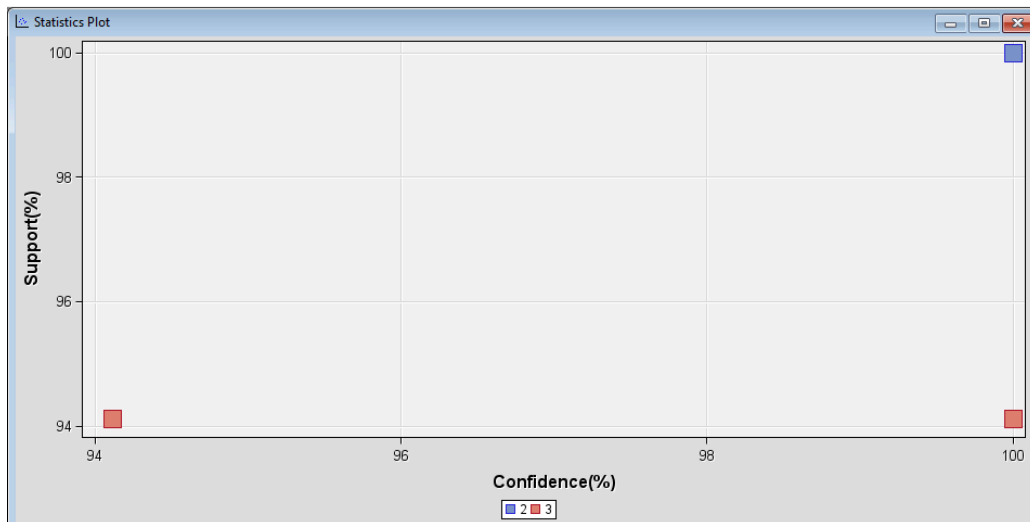
Diagram 7.1 below shows that by using the variables of course and target, there are a total of 12 combinations of rules that can be formed.



Map	Rule
RULE1	2 ==> 1
RULE2	3 ==> 1
RULE3	3 & 2 ==> 1
RULE4	1 ==> 2
RULE5	3 ==> 2
RULE6	3 & 1 ==> 2
RULE7	3 ==> 2 & 1
RULE8	2 ==> 3
RULE9	1 ==> 3
RULE10	2 & 1 ==> 3
RULE11	2 ==> 3 & 1
RULE12	1 ==> 3 & 2

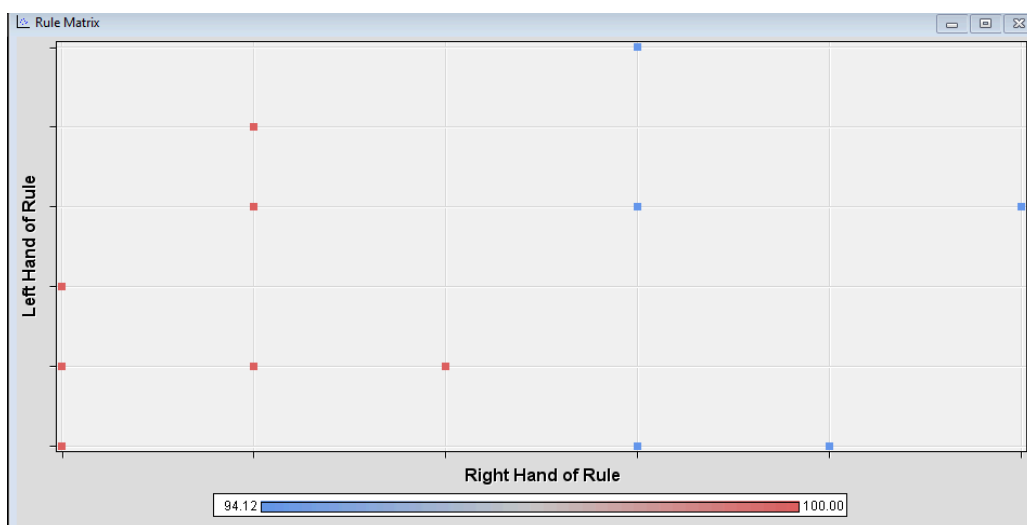
**Diagram 7.1 Association Rules Description**

Diagram 7.2 below shows the association rules statistics plot. The X-axis shows the confidence of the rule, while the Y-axis shows the support of the rule. Each point on the graph represents a single association rule. There is a rule at the top right-corner of the graph that represents a strong rule with both high support and high confidence. This is often the most interesting and reliable rule. However, the rule which is located at the bottom-left corner has the lowest support and confidence, indicating a less reliable rule.



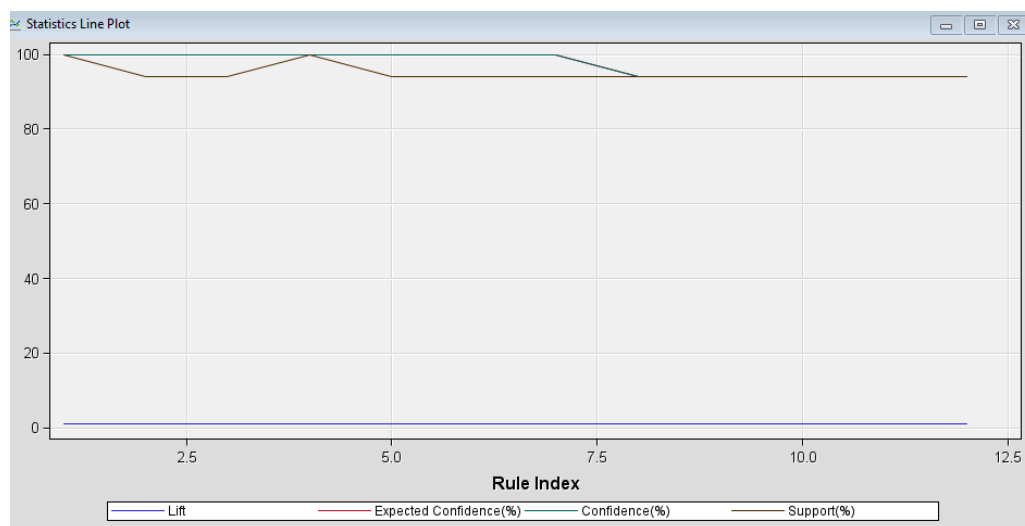
**Diagram 7.2 Association Rules Statistics Plot**

Diagram 7.3 below shows the rule matrix. The y-axis (left hand of rule) is used to describe and analyse the antecedent part of association rules while the x-axis (right-hand of rule) is used to describe and analyse the consequent part of association rules. This rule matrix allows us to understand the patterns and relationships between the conditions on the left-hand side and the outcomes on the right-hand side. According to the result shown, The rule items in the left hand of the rule consist of 2, 3, 3&2, 1, 3&1 and 2&1. in the right hand of the rule consists of 1, 2, 2&1, 3, 3&1 and 3&2.



**Diagram 7.3 Association Rules Matrix**

Diagram 7.4 below shows the statistics line plot. From the diagram below we can notice that the blue line (lift) for all the rules are consistent which is 1.0. This implies that the antecedent and consequent are independent, and the rule is not providing any additional information. Next, the green line (confidence) shows a consistent value which is 100% at the beginning but ultimately drops to 94.12 %. This indicates that in the beginning the presence of the antecedent guarantees the presence of the consequent. However, there are rules with a confidence of 94.12 % which implies that in 94.2% of the transactions containing the antecedent, the consequent is also present. Lastly, the brown line represents the support percentage. Although the support level of the rules might vary, the overall percentage for support is still considered high, which is greater than 94.2%. This indicates that the rules are present in a large proportion of transactions.



**Diagram 7.4 Association Rules Statistics Line Plot**

Diagram 7.5 below shows the rules table. It provides more detailed information about every rules.

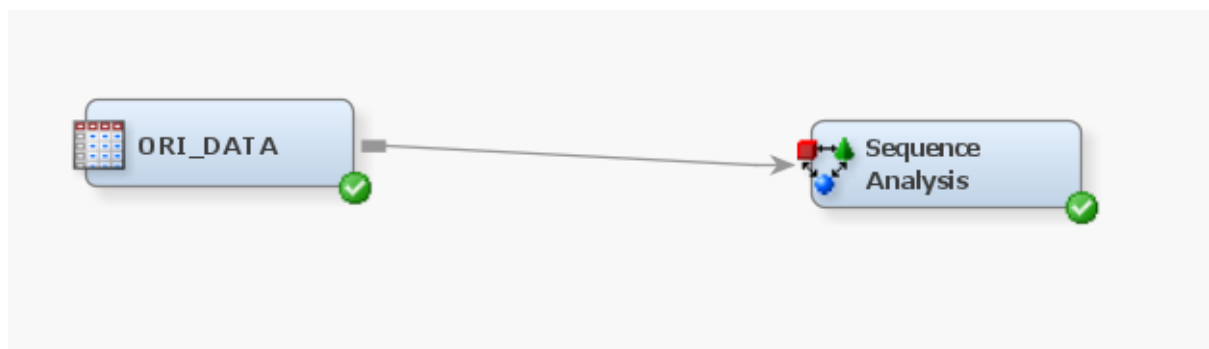
Relations	Expected Confidence(%)	Confidence(%)	Support(%)	Lift	Transaction Count	Rule	Left Hand of Rule	Right Hand of Rule	Rule Item 1	Rule Item 2	Rule Item 3	Rule Item 4	Rule Item 5	Rule Index	Transpose Rule
2	100.00	100.00	100.00	1.00	17.002	=> 1	2	1	2	=====	1			1	1
2	100.00	100.00	94.12	1.00	16.003	=> 1	3	1	3	=====	1			2	1
3	100.00	100.00	94.12	1.00	16.003 & 2	=> 1	3 & 2	1	3	2	=====	1		3	1
2	100.00	100.00	100.00	1.00	17.001	=> 2	1	2	1	=====	2			4	1
2	100.00	100.00	94.12	1.00	16.003	=> 2	3	2	3	=====	2			5	1
3	100.00	100.00	94.12	1.00	16.003 & 1	=> 2	3 & 1	2	3	1	=====	2		6	1
3	100.00	100.00	94.12	1.00	16.003	=> 2 & 1	3	2 & 1	3	=====	2	1		7	1
2	94.12	94.12	94.12	1.00	16.002	=> 3	2	3	2	=====	3			8	1
2	94.12	94.12	94.12	1.00	16.004	=> 3	1	3	1	=====	3			9	1
3	94.12	94.12	94.12	1.00	16.002 & 1	=> 3	2 & 1	3	2	1	=====	3		10	1
3	94.12	94.12	94.12	1.00	16.002	=> 3 & 1	2	3 & 1	2	=====	3	1		11	1
3	94.12	94.12	94.12	1.00	16.001	=> 3 & 2	1	3 & 2	1	=====	3	2		12	1

**Diagram 7.5 Association Rules Table**

## 7.2 Sequence Analysis

The purpose of sequence analysis in data mining is to discover interesting patterns and correlations in sequential data sets (Rithika, 2023). Sequence analysis in this context helps transform raw data into actionable insights.

To run the sequence analysis for our dataset, we first change the role of the dataset to Transaction and appoint “Course” as the ID. Then, we add an Association node to the diagram workspace and connect it to the dataset node. The Association node is renamed as Sequence Analysis.



**Diagram 7.6 Sequence Analysis Diagram**

Association Report														
Relations	Expected Confidence (%)	Confidence (%)	Support (%)	Lift	Transaction Count	Rule	Left Hand of Rule	Right Hand of Rule	Rule Item 1	Rule Item 2	Rule Item 3	Rule Item 4	Rule Item 5	Rule Index
2	100.00	100.00	100.00	1.00	17.00	2 ==> 1	2	1	2	=====>	1			1
2	100.00	100.00	94.12	1.00	16.00	3 ==> 1	3	1	3	=====>	1			2
3	100.00	100.00	94.12	1.00	16.00	3 < 2 ==> 1	3 < 2	1	3		2	=====>	1	3
2	100.00	100.00	100.00	1.00	17.00	1 ==> 2	1	2	1	=====>	2			4
2	100.00	100.00	94.12	1.00	16.00	3 ==> 2	3	2	3	=====>	2			5
3	100.00	100.00	94.12	1.00	16.00	3 < 1 ==> 2	3 < 1	2	3		1	=====>	2	6
3	100.00	100.00	94.12	1.00	16.00	3 ==> 2 < 1	3	2 < 1	3	=====>	2	1		7
2	94.12	94.12	94.12	1.00	16.00	2 ==> 3	2	3	2	=====>	3			8
2	94.12	94.12	94.12	1.00	16.00	1 ==> 3	1	3	1	=====>	3			9
3	94.12	94.12	94.12	1.00	16.00	2 < 1 ==> 3	2 < 1	3	2		1	=====>	3	10
3	94.12	94.12	94.12	1.00	16.00	2 ==> 3 < 1	2	3 < 1	2	=====>	3	1		11
3	94.12	94.12	94.12	1.00	16.00	1 ==> 3 < 2	1	3 < 2	1	=====>	3	2		12

**Diagram 7.7 Association Report**

Diagram 7.7 above shows the details of association rules.

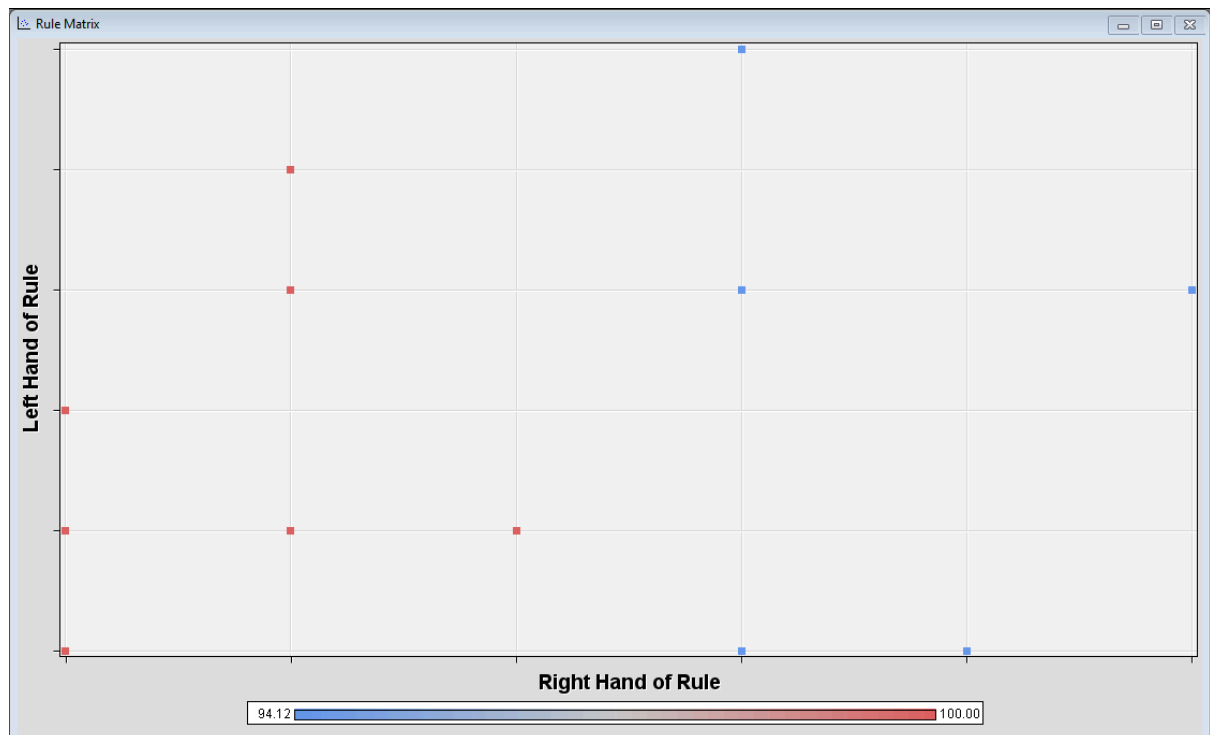


Diagram 7.8 Rule Matrix

Diagram 7.8 above shows the Rule Matrix scatter plot to visualize the relationship between the left-hand side (LHS) and the right-hand side (RHS) of association rules.



Diagram 7.9 Statistics Line Plot

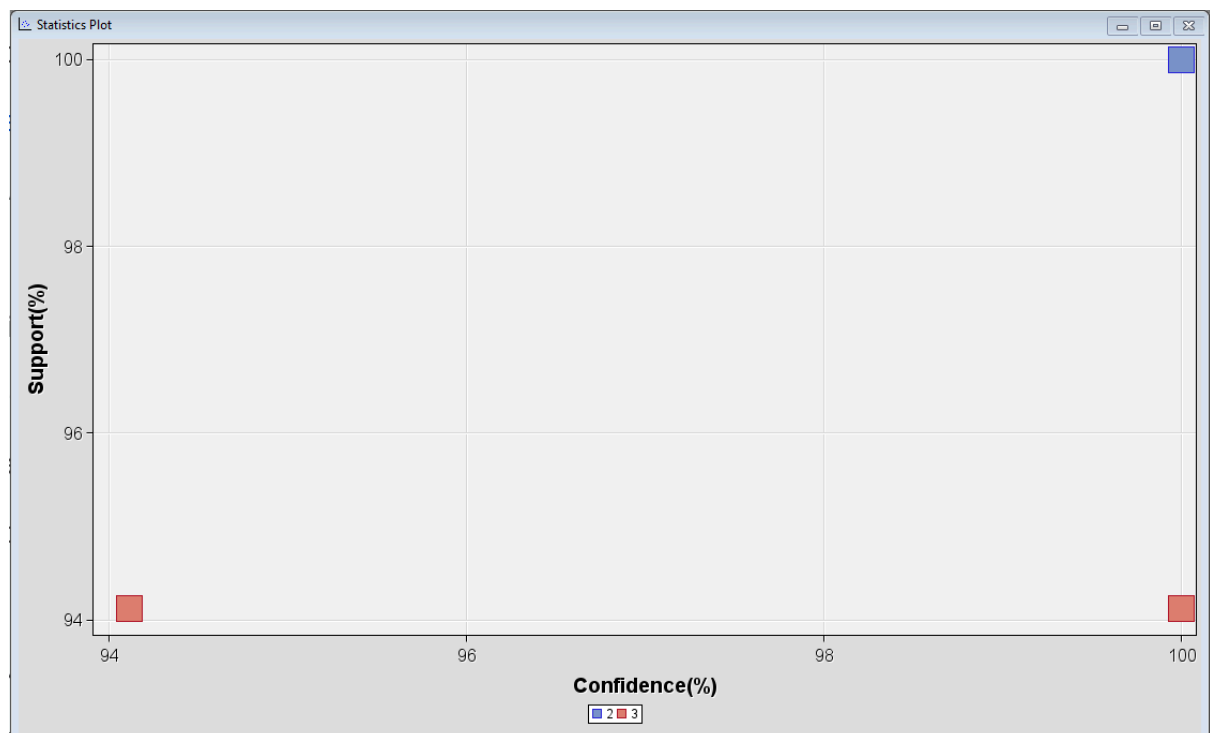
# Rule Statistics

The MEANS Procedure

Variable	Label	Minimum	Maximum	Mean
EXP_CONF	Expected Confidence(%)	94.1176471	100.0000000	97.5490196
CONF	Confidence(%)	94.1176471	100.0000000	97.5490196
SUPPORT	Support(%)	94.1176471	100.0000000	95.0980392
LIFT	Lift	1.0000000	1.0000000	1.0000000

**Diagram 7.10 Rule Statistics**

Diagram 7.9 and 7.10 above shows an overview of the association rule performance within the dataset. Both Expected Confidence and Confidence values are high, with an average of 97.549%. This suggests that the identified rules are highly reliable. Besides, the Support metric averages at 95.08% and all the lift values are at 1. In short, the high confidence and support imply that the rules are meaningful and can be beneficial for predictive analytics and decision-making processes.



**Diagram 7.11 Statistics Plot**



## Sequence Report

The FREQ Procedure

Relations				
SET_SIZE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
-----				
2	6	50.00	6	50.00
3	6	50.00	12	100.00

**Diagram 7.12 Sequence Report**

According to Diagram 7.11 and 7.12, there are two set sizes, 2 and 3. Each of these set sizes accounts for exactly half of the itemsets, with a frequency and percent of 50%.

## 7.3 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

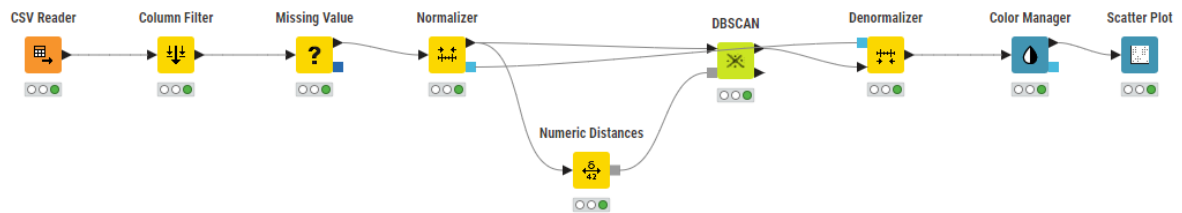


Diagram 7.13 DBSCAN diagram

### Correlation between Courses and Average Grades: A Scatter Plot Analysis

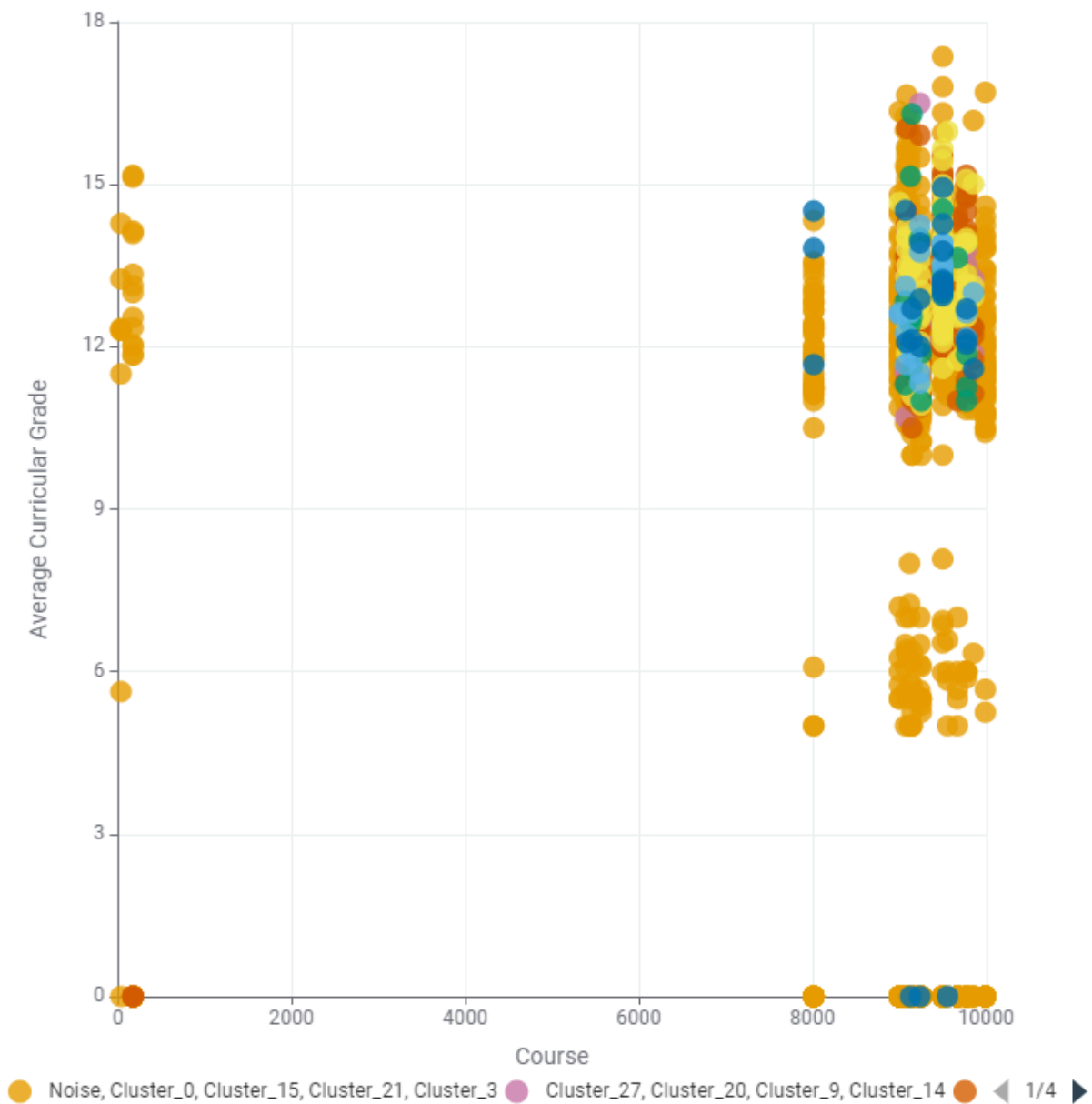


Diagram 7.14 Scatter Plot Diagram

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm used for data analysis and pattern recognition. It groups the data point based on their density, identifying the clusters of high-density regions and classifying outliers as noise (*Sharma, 2023*). Knime is used to implement DBSCAN on the chosen dataset.

To initiate the process, CSV Reader Node extracts data from a CSV file, and the subsequent output flows into the Column Filter Node. Then, the Missing Value Node handles the missing values within the dataset. The processed data is then converged into the Normalizer Node, normalizing the data, potentially scaling it to a standard range. Subsequently, the normalized table links to the Numeric Distances Node, where numeric distances between data points are computed.

DBSCAN Node is then implemented on the normalized data, facilitating the identification of clusters. The Epsilon value of DBSCAN Node is changed to 0.6. With this choice, the algorithm considers data points within a radius of 0.6 units to be potential neighbours. Smaller values may lead to tighter and more compact clusters. The choice of 0.6 reflects the sensitivity of the algorithm to local density variations and impacts the overall structure of the identified clusters in the dataset.

The normalized model is then connected with the Denormalizer Node, ensuring the transformed data returns to its original scale. Simultaneously, the output of the Numeric Distance Node feeds into the DBSCAN's distance model port, contributing crucial information. The data with cluster IDs generated from DBSCAN is then connected to the Denormalizer Node. The denormalized output then integrates with the Color Manager Node, to manage colour for an effective visualization. The table with the color information is generated and connected to the Scatter Plot Node to generate a scatter plot based on the clustered data.

In the Scatter Plot Node, course is chosen as the horizontal dimension while AverageCurricularGarde is chosen as the vertical dimension. The addition of colour-coded clusters further enhances our ability to discern patterns within the chosen dataset and also allows us to differentiate between noise and meaningful clusters. As we explore the scatter plot, there is a tight gathering of points between

8000 and 10000 on the x-axis and 9 - 15 on the y-axis. This suggests that there is a significant portion of the dataset consistently achieving grades within the 9 - 15 range. While for the outliers and points, they are located in the less densely populated regions that contribute diversity to the academic landscape within the dataset. In short, the scatter plot allows us to identify the richness and diversity of academic performance patterns, paving the way for the targeted strategies to enhance educational outcomes.

## 8.0 Model

Predictive modelling is one of the data mining technologies that works by creating a model to assist in forecasting future events by analysing both historical and present data. Models are built from historical event records and are used to predict future occurrences of these events. The results of the predictive models are called predictions. Predictions is the best guess for the target of the given set of input measurements based on the patterns learned by studying the training data. We have chosen four models which are decision tree, regression, gradient boosting and random forest. The figure below shows our model diagram.

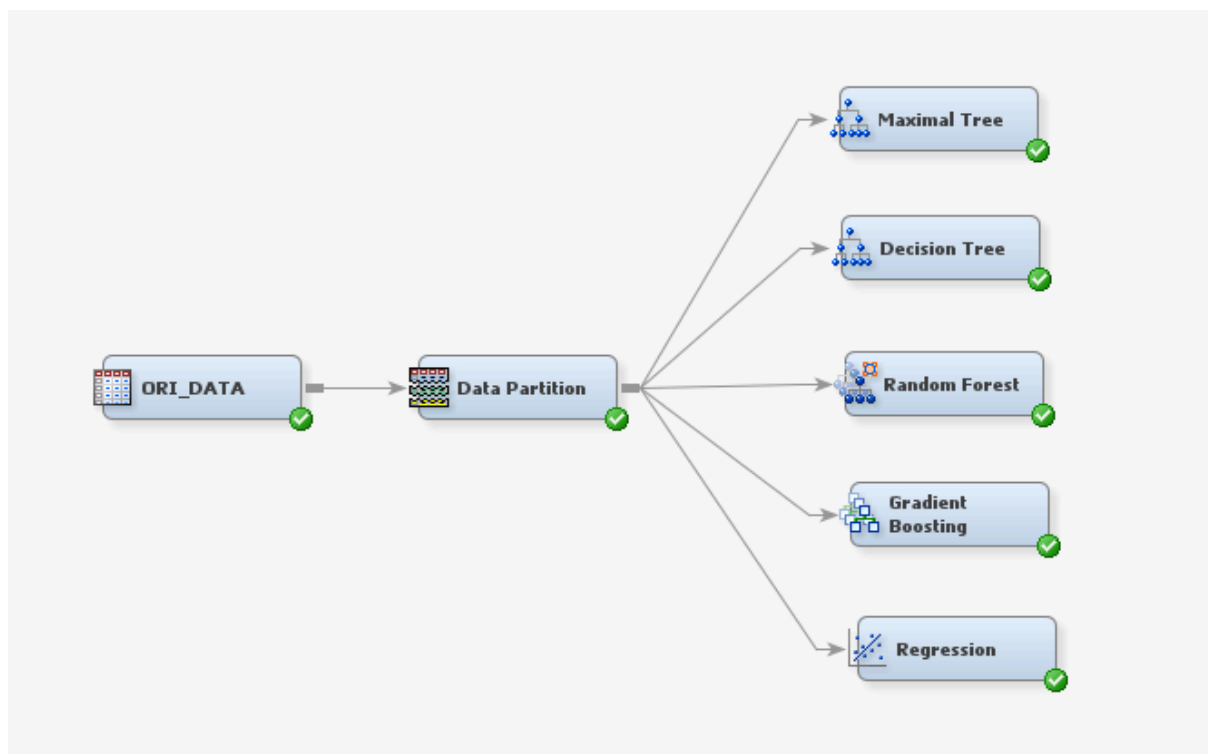


Figure 8.0 Model Diagram

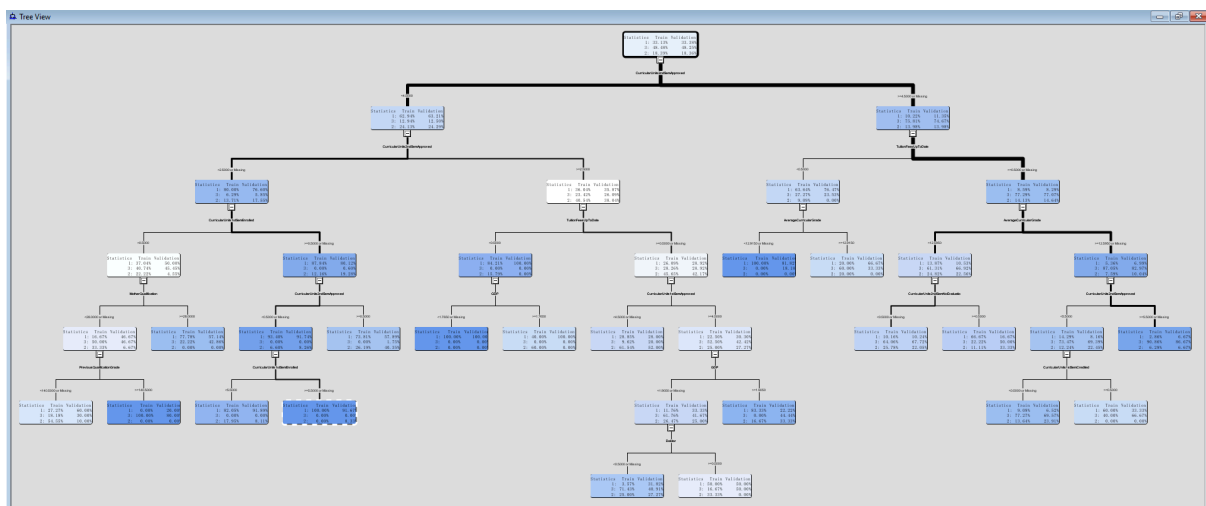
### 8.1 Decision Tree

The first machine learning model we chose is the decision tree. Decision trees have hierarchical tree structure which includes the root node, internal nodes and leaf nodes. It is a class of supervised learning in data mining techniques that separate a huge collection of heterogeneous records into smaller groups of homogenous records by applying the directed knowledge discovery (Ghoshon, A. M. , 2011).

We will start the modelling by cultivating decision trees using **split search**. The split search will start by selecting an input for partitioning the available training data. For a selected split point, two groups will be generated which are the branch left where the input value is lesser than the split point and the branch right where the input has larger values compared to the split point. In our case, we select the partition with the highest logworth as the split input. This partitioning process will be repeated for every input in the training data.

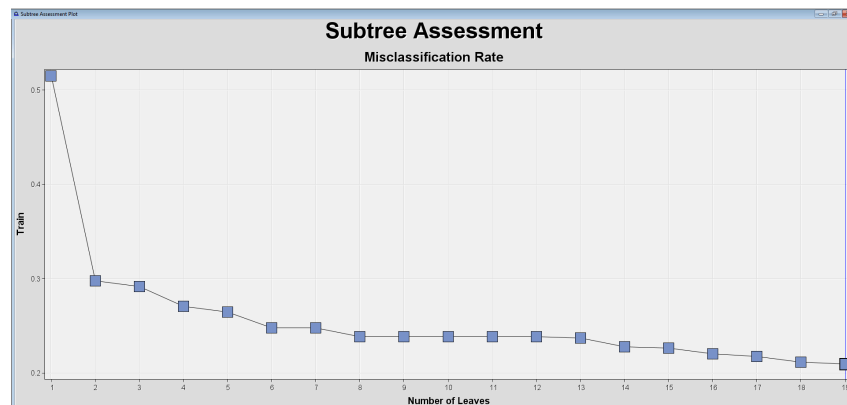
In our case, the `CurricularUnits2ndSemApproved` input is first selected to be splitted with the highest log worth of **59.0071**. As a result, the training data is partitioned into two subsets. The data for approved second semester curricular units which are **less than 4.5** will be selected to the left branch, having the highest percentage of **Target = 1** for both test and validation data (62.94% and 63.91% respectively). On the other hand, the right branch has the highest percentage of **Target = 3** for both test and validation data (75.81% and 74.67% respectively).

Then we will perform splitting for the left branch and right branch respectively using the same method. After that, we will use the train node function on the root node to train the model and perform splitting to grow a tree until stopping rules prohibit further growth to generate a **maximal tree**. *Figure 7.2* shows the tree generated.



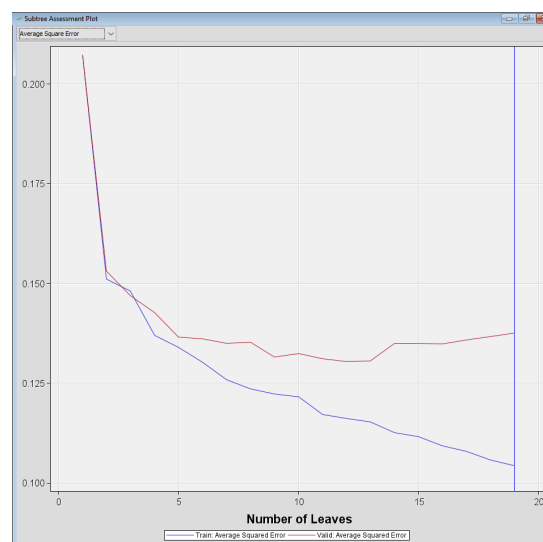
**Figure 8.1.1 Decision Tree (Maximal Tree)**

There are **36** nodes in the decision tree in *Figure 7.2*. The decision tree has 2 branches with a **tree depth of 6**. The gradient-coloured-nodes indicate the percentage of correctly classified observations where light colour represents high percentage while dark colour represents low percentage.



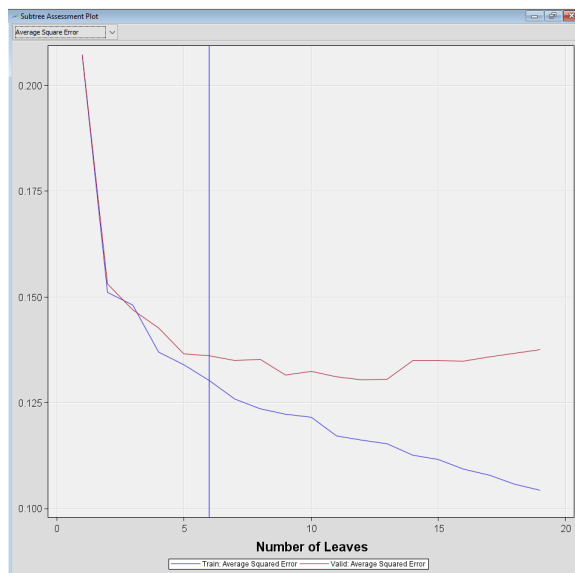
**Figure 8.1.2: Assessment Plot (Maximal Tree)**

From the assessment plot shown in *Figure 7.3*, the majority of the improvements occur on the first split. Besides, the maximal 19-leaf-tree generates the lowest average square error. This seems to suggest that the maximal tree is preferred. To use a maximal tree as a model, we set the **Use Frozen Tree** property to Yes to prevent the maximal tree from being manipulated by other property settings when the flow runs. After running the maximal tree node, we get the average square error plot as below.



**Figure 8.1.3 Average Square Error Plot (Maximal Tree)**

This graph is identical to the one which is generated in the interactive decision tree tool. However, we can see that the performance of the validation sample only improves up to a tree of approximately 5 leaves and becomes worse as the model complexity increases. This is an example of model overfitting where the model has accurate predictions for test data but not valid data.



**Figure 8.1.4 Average Square Error Plot  
(Decision Tree)**

From the above figure, we can see that the **6-leaf tree** has the lowest value of average squared error on the validation sample. The tree that is generated by the decision tree node also shows that the optimal tree consists of 6 leaves.



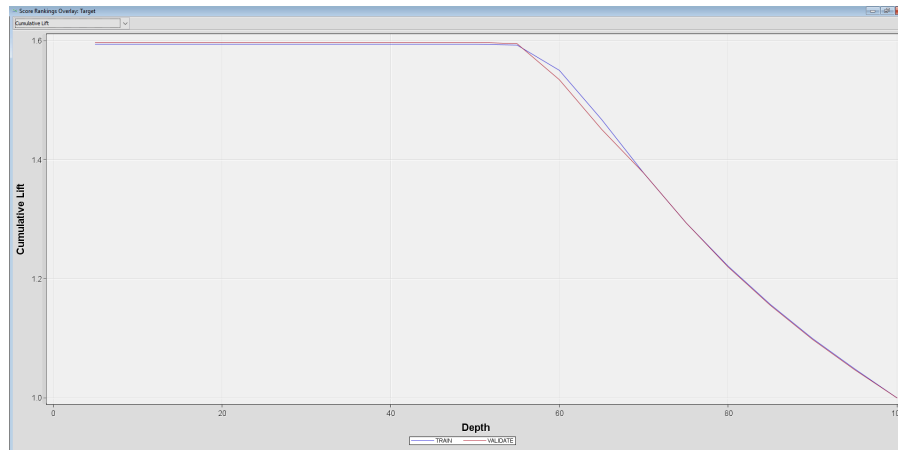


Figure 8.1.6 Score Ranking Overlays (Decision Tree)

Depth	Gain	Lift	Cumulative Lift	% Response	Cumulative % Response	Number of Observations	Mean Posterior Probability
5	59.4161	1.59416	1.59416	77.2853	77.2853	33	0.77285
10	59.4161	1.59416	1.59416	77.2853	77.2853	33	0.77285
15	59.4161	1.59416	1.59416	77.2853	77.2853	33	0.77285
20	59.4161	1.59416	1.59416	77.2853	77.2853	33	0.77285
25	59.4161	1.59416	1.59416	77.2853	77.2853	33	0.77285
30	59.4161	1.59416	1.59416	77.2853	77.2853	33	0.77285
35	59.4161	1.59416	1.59416	77.2853	77.2853	33	0.77285
40	59.4161	1.59416	1.59416	77.2853	77.2853	33	0.77285
45	59.4161	1.59416	1.59416	77.2853	77.2853	33	0.77285
50	59.4161	1.59416	1.59416	77.2853	77.2853	32	0.77285
55	59.2749	1.57867	1.59275	76.5342	77.2169	33	0.76534
60	55.0155	1.08292	1.55016	52.5000	75.1519	33	0.52500
65	46.7687	0.48057	1.46769	23.2984	71.1538	33	0.23298
70	37.6823	0.19834	1.37682	9.6154	66.7487	33	0.09615
75	29.3927	0.13590	1.29393	6.5884	62.7299	33	0.06588
80	22.1022	0.12966	1.22102	6.2857	59.1954	33	0.06286
85	15.6709	0.12966	1.15671	6.2857	56.0776	33	0.06286
90	9.9555	0.12966	1.09955	6.2857	53.3067	33	0.06286
95	4.8426	0.12966	1.04843	6.2857	50.8279	33	0.06286
100	0.0000	0.05267	1.00000	2.5536	48.4802	32	0.02554

Figure 8.1.7 Output (Training Set)

Depth	Gain	Lift	Cumulative Lift	% Response	Cumulative % Response	Number of Observations	Mean Posterior Probability
5	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
10	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
15	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
20	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
25	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
30	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
35	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
40	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
45	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
50	59.7180	1.59718	1.59718	77.0718	77.0718	33	0.77285
55	59.6202	1.57542	1.59520	76.0219	76.9764	33	0.76534
60	53.4544	0.86730	1.53454	41.8517	74.0493	33	0.51736
65	45.1112	0.44993	1.45111	21.7112	70.0233	33	0.18177
70	37.7066	0.41447	1.37707	20.0000	66.4502	33	0.09615
75	29.3345	0.12125	1.29334	5.8511	62.4103	33	0.06286
80	22.0089	0.12125	1.22009	5.8511	58.8753	33	0.06286
85	15.5452	0.12125	1.15545	5.8511	55.7562	33	0.06286
90	9.7996	0.12125	1.09800	5.8511	52.9837	33	0.06286
95	4.6589	0.12125	1.04659	5.8511	50.5031	33	0.06286
100	0.0000	0.08715	1.00000	4.2055	48.2549	32	0.04518

Figure 8.1.8 Output (Validation Set)

We also analyse the cumulative lift which indicates how well a model is performing compared to a random baseline. From *Figure 8.1.6*, the first 50% of cases, the training and validation lift is approximately 1.6, and then decline gradually as moving down the depth axis. This indicates that **the model becomes less effective when the decision tree depth goes deeper**.

## 8.2 Regression

Regression is used for predicting a numeric target variable based on one or more predictor variables. Continuous and discrete inputs are both available to use. A regression model is able to show whether changes observed in the dependent variable are associated with changes in one or more of the explanatory variables.

Firstly, we will run the regression node without any extra configuration. The default regression type used by SAS enterprise miner is **Logistic Regression** with **Logit** link function. Logistic regression, also known as the sigmoid function) is used for binary classification problems, where the outcome variable is dichotomous, meaning it has only two possible outcomes (usually coded as 0 and 1). Besides, we are implementing a sequential selection method in the Regression node by setting the Selection Model to **Stepwise**. Sequential selection improves the model's performance and finds a subset of variables that best explains the variation in target variable.

Summary of Stepwise Selection									
Step	Entered	Effect	Removed	DF	Number	Score	Wald	Pr > ChiSq	Validation
					In	Chi-Square	Chi-Square		Error Rate
1	CurricularUnits2ndSemApproved			2	1	262.8973		<.0001	1053.1
2	CurricularUnits2ndSemEnrolled			2	2	148.8320		<.0001	946.4
3	TuitionFeesUpToDate			2	3	47.2663		<.0001	876.8
4	AgeAtEnrollment			2	4	24.3491		<.0001	890.8
5	CurricularUnits1stSemApproved			2	5	19.1507		<.0001	874.6
6	CurricularUnits1stSemCredited			2	6	15.6976		0.0004	861.5
7	ScholarshipHolder			2	7	10.2562		0.0059	857.1
8	Gender			2	8	8.6783		0.0130	855.2
9	CurricularUnits2ndSemEvaluations			2	9	7.4784		0.0238	846.9
10	MotherOccupation			2	10	6.9736		0.0306	850.2
11		MotherOccupation		2	9		5.4080	0.0669	846.9

**Figure 8.2.1 Output (Regression)**

From the output in Figure 8.2.1, all the variables entered are **statistically significant** (p-values < 0.05) except for `MotherOccupation` which means that it does not provide significant explanatory power beyond the other variables. Thus, this input is removed. The order of variable entry suggests their relative importance in predicting outcome variables. `CurricularUnits2ndSemApproves` was the most important predictor, followed by `CurricularUnits2ndSemEnrolled` and so on.

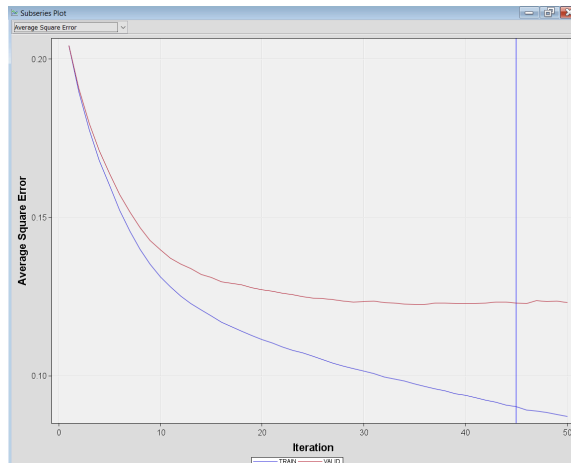
Odds Ratio Estimates		
Effect	Target	Point Estimate
AgeAtEnrollment	3	0.939
AgeAtEnrollment	2	0.919
CurricularUnits1stSemApproved	3	1.901
CurricularUnits1stSemApproved	2	1.023
CurricularUnits1stSemCredited	3	0.761
CurricularUnits1stSemCredited	2	0.882
CurricularUnits2ndSemApproved	3	3.308
CurricularUnits2ndSemApproved	2	1.642
CurricularUnits2ndSemEnrolled	3	0.276
CurricularUnits2ndSemEnrolled	2	0.631
CurricularUnits2ndSemEvaluations	3	0.994
CurricularUnits2ndSemEvaluations	2	1.115
Gender	3	0.390
Gender	2	0.669
ScholarshipHolder	3	2.605
ScholarshipHolder	2	1.080
TuitionFeesUpToDate	3	46.915
TuitionFeesUpToDate	2	10.215

**Figure 8.2.2 Odds Ratio Estimate (Regression)**

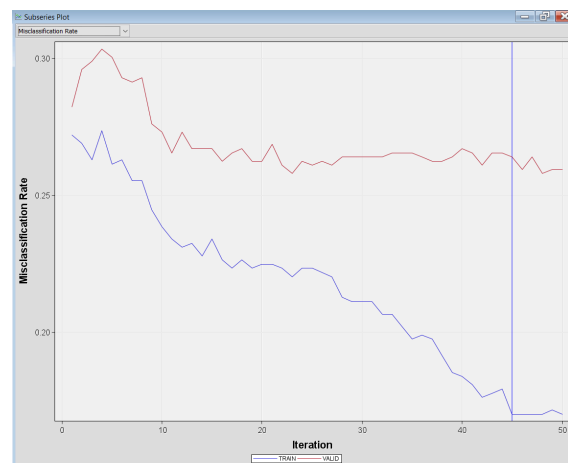
The **odds ratio** is a measure used in logistic regression to quantify the relationship between a binary outcome variable and one or more predictor variables. It is a way to express the odds of an event happening in one group compared to the odds of it happening to another group. For example, according to *Figure 8.2.2*, the chances for Target 2 and 3 to get a ‘Yes’ answer decreases when the `AgeAtEnrollment` increases because its point estimate is less than 1. On the other hand, the chances for Target 2 and 3 to get a ‘Yes’ answer increases when the `CurricularUnits1stSemApproved` increases because its point estimate is more than one.

### 8.3 Gradient Boosting

Gradient boosting is a method that is able to predict efficiently and accurately with large and complex datasets. It involves finding the best way to partition the data based on a single variable. It is aimed to create segments where the target variable is distributed in a way that makes them more similar within the segments. The process repeats by further dividing each segment until an optimal partition is achieved. These partitions are then combined to create a predictive model.



**Figure 8.3.1 Average Square Error  
(Gradient Boosting)**



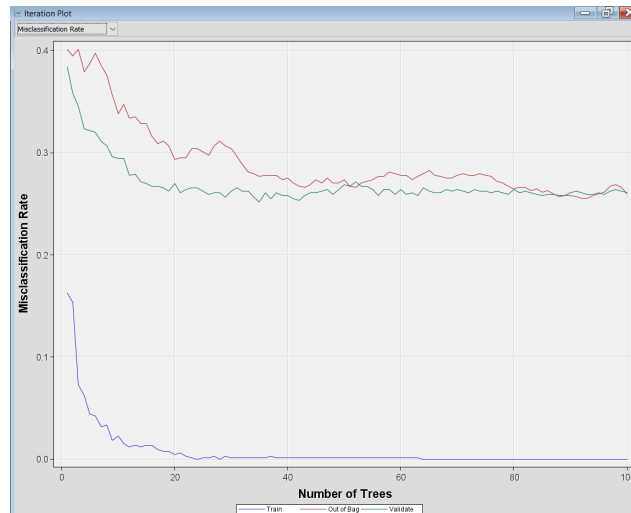
**Figure 8.3.2 Misclassification Rate  
(Gradient Boosting)**

According to *Figure 8.3.1*, the average square error decreases significantly when the tree depth is less than 10. As the depth increases the model stills perform well in the train partition and achieve near 0 average square error when the depth is 0. However, the model does not show much improvement on the validation set data after the depth of 10 and its performance regresses as the depth increases. This leads to the suggestion that we should use a tree with smaller depth in order to predict the target accurately.

Besides, by analysing *Figure 8.3.2*, a downward trend in misclassification rate of the test data suggests improvement as depth of tree increases. However, there is a significant divergence between the training and validation lines. When the training error decreases, the validation error increases. This suggests that the model is undergoing overfitting. To overcome this problem, we should find the optimal depth of tree which stops the iteration before overfitting occurs.

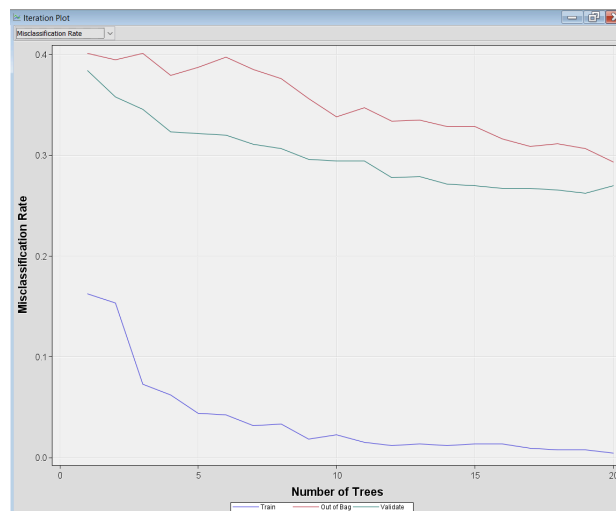
## 8.4 Random Forest

Random forest is a model that is made up of a collection of decision trees. It combines the results of each tree in the collection to make more accurate predictions and reduce overfitting.



**Figure 8.4.1 Misclassification Rate 100 Trees (Random Forest)**

According to the random forest results of misclassification rate plot, the red line indicates the standard deviation of the out-of-bags (OOB) error rate across the trees in the forest. The OOB error rate is an estimate of the model's generalisation error or how well it will perform on unseen data. The general trend of OOB in the plot decreases as the number of trees increases. This is because each tree in a random forest makes its own prediction, and by averaging the predictions of all the trees, we can get a more accurate prediction. However, there is also a point of diminishing returns, where adding more trees doesn't improve the model's accuracy, which is 20 trees as we can see that the train data misclassification plot started to flatten. Therefore, we can set 20 as the maximum number of trees in the forest and run the random forest node again.



**Figure 8.4.2 Misclassification Rate 20 Trees (Random Forest)**

Variable Name	Number of Splitting Rules	Train: Gini Reduction	Train: Margin Reduction	OOB: Gini Reduction	OOB: Margin Reduction	Valid: Gini Reduction	Valid: Margin Reduction	Label
PreviousQualificationGrade	192	0.031220	0.056373	-0.02669	-0.00684	-0.02966	-0.00741	
CurricularUnits2ndSemGrade	143	0.051488	0.082179	-0.00240	0.02881	0.00711	0.04198	
MotherQualification	126	0.018950	0.034727	-0.02709	-0.01213	-0.02100	-0.00640	
MotherOccupation	124	0.022558	0.038568	-0.01259	0.00506	-0.02148	-0.00461	
UnemploymentRate	116	0.018140	0.032670	-0.01713	-0.00364	-0.01869	-0.00496	
CurricularUnits1stSemGrade	109	0.055726	0.094674	0.01834	0.05518	0.00333	0.04189	
FatherQualification	109	0.017852	0.032089	-0.02068	-0.00763	-0.01945	-0.00563	
InflationRate	99	0.017596	0.030237	-0.02344	-0.00771	-0.02215	-0.00893	
CurricularUnits2ndSemAppro...	98	0.061735	0.097449	0.02738	0.06315	0.03431	0.06976	
GDP	97	0.015998	0.027058	-0.01864	-0.00708	-0.01901	-0.00714	
AverageCurricularGrade	96	0.036939	0.057132	-0.00637	0.01451	0.00224	0.02199	
FatherOccupation	92	0.016375	0.027093	-0.01880	-0.00537	-0.01888	-0.00716	
CurricularUnits2ndSemEvalu...	87	0.017700	0.028650	-0.01375	-0.00051	-0.01110	0.00072	
AgeAtEnrollment	81	0.022259	0.037853	-0.00579	0.00863	-0.01211	0.00352	
CurricularUnits1stSemEvalu...	73	0.014565	0.024080	-0.01484	-0.00242	-0.01292	-0.00042	
ApplicationMode	72	0.016947	0.027357	-0.01397	-0.00452	-0.00985	0.00104	
AdmissionGrade	68	0.016091	0.027682	-0.01732	-0.00533	-0.01266	-0.00260	
Course	68	0.013471	0.022085	-0.01014	-0.00129	-0.00780	0.00098	
CurricularUnits1stSemEnrolled	66	0.017571	0.028293	-0.00120	0.01012	-0.00183	0.00890	
CurricularUnits1stSemAppro...	62	0.060712	0.091967	0.04180	0.07315	0.04144	0.07355	
CurricularUnits2ndSemEnroll...	47	0.009694	0.016403	-0.00250	0.00410	-0.00390	0.00342	
TuitionFeesUpToDate	46	0.028061	0.045459	0.02343	0.04152	0.01875	0.03528	
Gender	42	0.008569	0.014864	-0.00232	0.00258	-0.00178	0.00333	
PreviousQualification	32	0.004839	0.008139	-0.00383	-0.00171	-0.00567	-0.00288	
ScholarshipHolder	28	0.005233	0.009536	-0.00203	0.00315	-0.00069	0.00324	
Debtor	21	0.004728	0.006545	-0.00015	0.00162	-0.00200	0.00030	
CurricularUnits2ndSemNoEv...	18	0.003679	0.005769	-0.00273	-0.00076	-0.00222	0.00021	
CurricularUnits2ndSemCredit...	14	0.002497	0.004566	-0.00226	-0.00063	-0.00235	-0.00050	
MaritalStatus	14	0.002376	0.004660	-0.00023	0.00169	-0.00140	0.00051	
Displaced	13	0.001405	0.002768	-0.00163	-0.00050	-0.00158	-0.00043	
CurricularUnits1stSemCredited	12	0.002145	0.003332	-0.00166	-0.00084	-0.00113	-0.00042	
DaytimeEveningAttendance	12	0.001704	0.002789	-0.00096	0.00034	-0.00195	-0.00121	
CurricularUnits1stSemNoEva...	11	0.001889	0.003504	-0.00154	-0.00023	-0.00151	-0.00010	
Nationality	5	0.000652	0.001183	-0.00281	-0.00172	-0.00044	-0.00019	
International	3	0.000255	0.000393	-0.00007	0.00015	-0.00007	0.00013	
EducationalSpecialNeeds	0	0.000000	0.000000	0.00000	0.00000	0.00000	0.00000	

**Figure 8.4.3 Variable Importance (Random Forest)**

After reducing the number of trees, we can see that the misclassification rate of the validation data decreases over time and is no longer undergoing overfitting because the plot is not flattened overtime.

Besides, *Figure 8.4.3* shows the variable importance. The `PreviousQualificationGrade` seems to be the most important variable in predicting the target as it has the highest number of splitting rules followed by the `CurricularUnits2ndSemGrade`, `MotherQualification` and so on.

## 9.0 Assess

To assess and compare the performance of various connected models, the Model Comparison node under Assess category in SAS Enterprise Miner is utilised. This node can review and compare the performance of the connected models with data mining measures for this project. (SAS Help Center, n.d.).

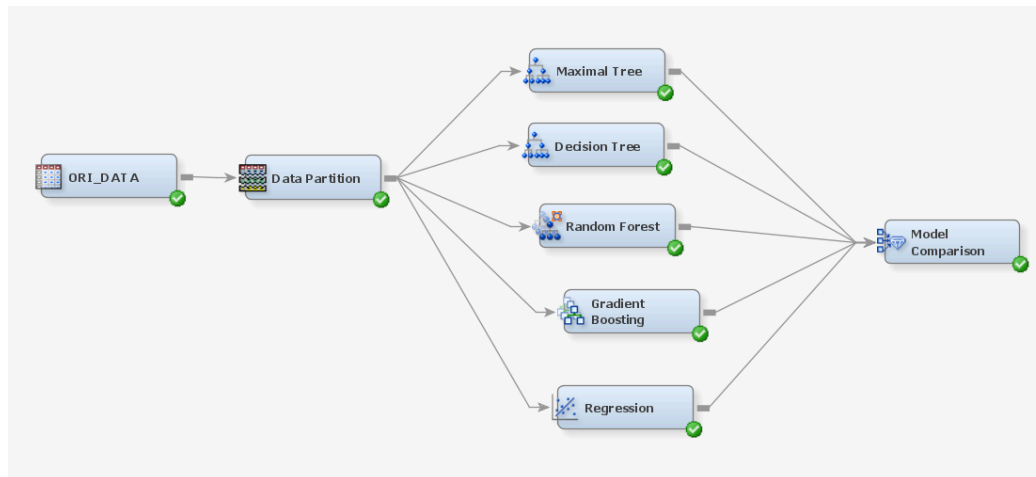


Diagram 9.1 Model Comparison

In our group project, the Model Comparison node is connected with the models to run the comparison analysis.

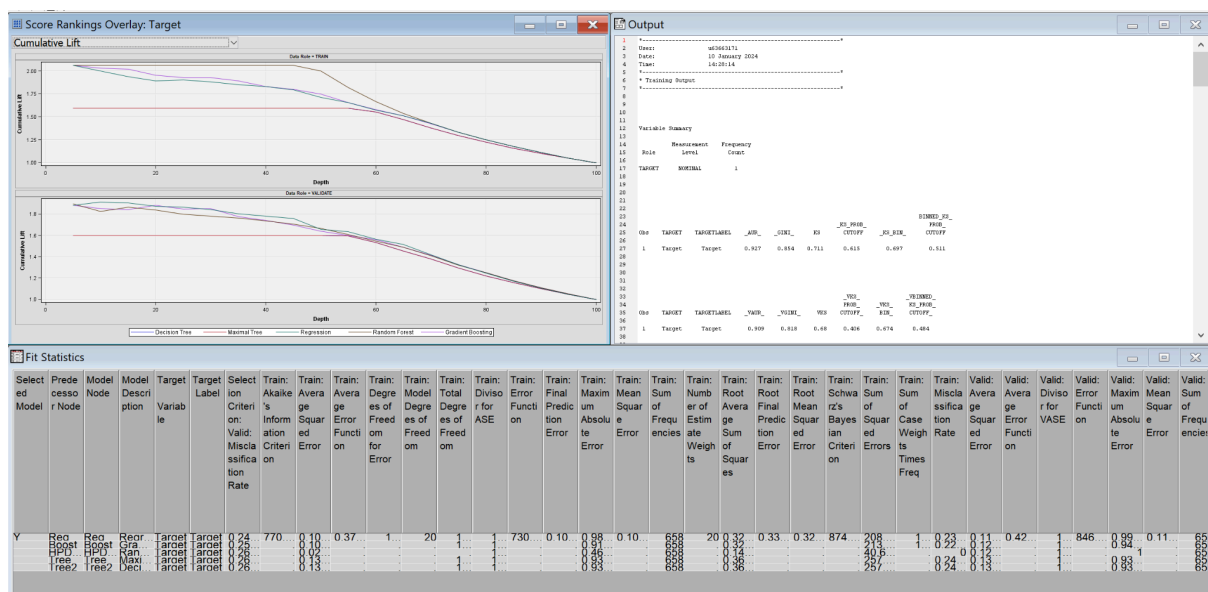
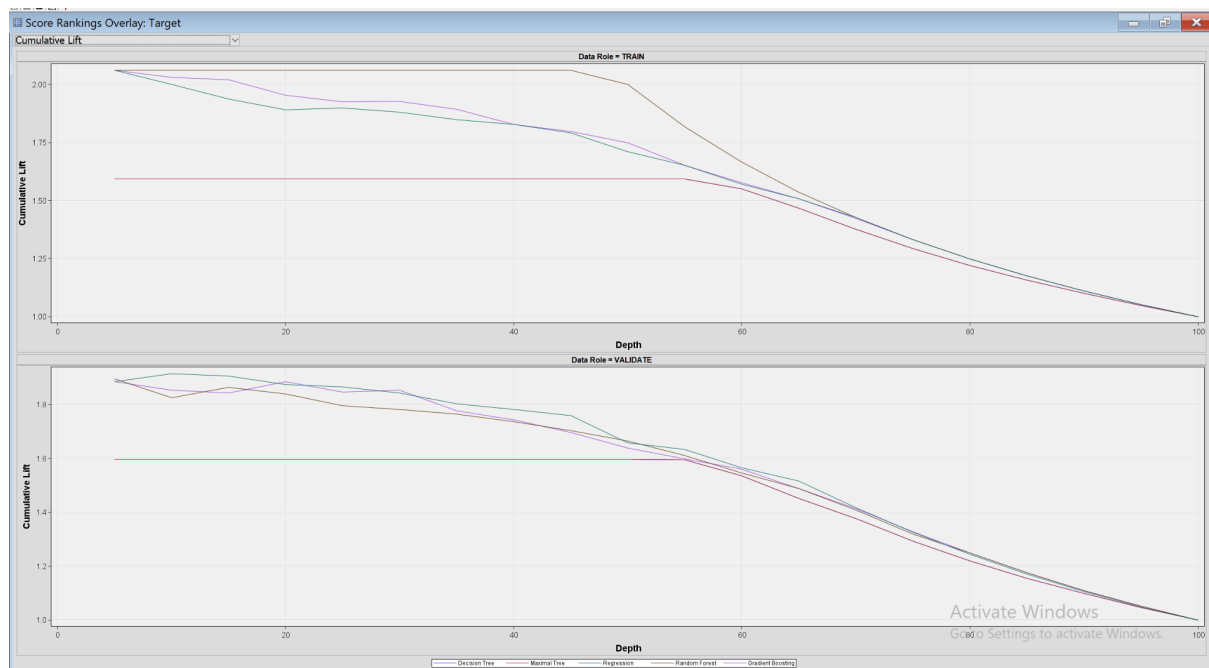


Diagram 9.2 Result of Model Comparison Node

Diagram 9.2 above shows the result after running the Model Comparison Node. The result shows 3 windows of comparison which are Score Rankings Overlay Charts by Target, Output Text File, and Fit Statistics Table. The details of each window are analysed below.



**Diagram 9.3 Score Rankings Overlay by Target, Cumulative Lift Charts**

Diagram 9.3 above shows the Cumulative Lift Charts, which evaluates the performance of various predictive models on both training and validation datasets.

The upper chart represents how well the model performs on the training dataset. First, the Random Forest model has the highest predictive lift, outperforming other models at lower depths. This means that it is able to identify the most probable positive outcomes in the smaller portions of the dataset. As the depth increases, its performance converges with that of the Gradient Boosting and Regression models, which suggests that despite different modelling approaches, their overall predictive capabilities become similar. Then, the Maximal Tree and Decision Tree models exhibit the same performance throughout all depths, implying that the additional complexity of the maximal, unpruned tree does not yield improved predictions over the pruned decision tree in this tracking scenario.



Next, the bottom chart shows the cumulative lift for validation data, which evaluates the performance of different predictive models on a dataset that they were not trained on. This helps us to understand how well a model generalises to new data. First, the Regression model starts off strong, indicating it is effective for the most confident predictions but experiences a gradual decrease in lift as more data is considered. Then, the Gradient Boosting and Random Forest models display a competitive performance, with the lift fluctuating between them across different depths. The crossing lines imply that neither consistently outperforms the other across the full range of the data. Next, the Maximal Tree and Decision Tree models are represented by the same line at the bottom, showing the least amount of lift. This indicates that they have a lower predictive capability on the validation set compared to the more complex models.

Fit Statistics																													
Select ed Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Select ion Criteria: Misclassification Rate	Train: Akaike's Information Criterion	Train: Average Squared Error	Train: Average Squared Error	Train: Degrees of Freedom for Error	Train: Model Degrees of Freedom	Train: Total Degrees of Freedom	Train: Divisor for ASE	Train: Error Function	Train: Final Prediction Error	Train: Maximum Absolute Error	Train: Mean Squared Error	Train: Sum of Squared Errors	Train: Number of Estimates	Train: Root Average Sum of Squares	Train: Root Final Prediction Error	Train: Root Mean Squared Error	Train: Schwarz's Bayesian Criterion	Train: Sum of Squared Errors	Train: Sum of Case Weights Times Freq	Train: Misclassification Rate	Valid: Misclassification Rate	Valid: Average Squared Error	Valid: Average Function
Y	Regression Model	Regression Model	Regression Model	Target	Target	0.24	770	0.10	0.37	1	20	1	1	730	0.10	0.98	0.10	658	20	0.32	0.33	0.32	874	208	1	0.23	0.11	0.42	
	Boosted Tree	Boosted Tree	Gradient Boosting	Target	Target	0.26		0.09								0.97		658		0.34				208		0.22	0.12		
	Random Forest	Random Forest	Random Forest	Target	Target	0.26		0.13								0.93		658		0.36				257		0.24	0.13		
	Maximal Tree	Maximal Tree	Maximal Tree	Target	Target	0.25																							
	Decision Tree	Decision Tree	Decision Tree	Target	Target	0.25																							

Diagram 9.4 Fit Statistics Table 1

Fit Statistics																														
Valid: Divisor for VASE	Valid: Error Function	Valid: Maximum Absolute Error	Valid: Mean Squared Error	Valid: Sum of Squared Errors	Valid: Root Mean Squared Error	Valid: Sum of Squared Errors	Valid: Sum of Case Weights Times Freq	Valid: Misclassification Rate	Train: Frequency of Classified Cases	Train: Number of Wrong Classifications	Valid: Frequency of Classified Cases	Valid: Number of Wrong Classifications	Train: ROC Index	Train: Gini Coefficient	Train: Kolmogorov-Smirnov Statistic	Train: Kolmogorov-Smirnov Probability Cutoff	Train: Binomial Two-Way Kolmogorov-Smirnov Probability Cutoff	Valid: ROC Index	Valid: Gini Coefficient	Valid: Kolmogorov-Smirnov Statistic	Valid: Kolmogorov-Smirnov Probability Cutoff	Valid: Binomial Two-Way Kolmogorov-Smirnov Probability Cutoff	Train: Gain	Valid: Gain	Train: Lift	Train: Cumulative Lift	Valid: Cumulative Lift	Train: Percent Response	Train: Cumulative Percent Response	
1	846	0.99	0.11	659	0.34	0.34	233	1	0.24				0	0	0	0	0	0	0	0.68	0	0	0	100	91.5	1.93	2.00	1.94	91	93.9
1		0.94		659	0.35		238	1	0.28				0	0	0	0	0	0	0	0.66	0	0	0	103	89.2	2.00	2.03	1.92	96	98.3
1		0.93		659	0.36		269		0.28	658	0	659	172	0	0	0	0	0	0.84	0.88	0	0	58.4	59.7	1.59	1.65	1.65	77.2	77.2	

Diagram 9.5 Fit Statistics Table 2

Valid: Lift	Valid: Cumulative Lift	Train: Percent Response	Train: Cumulative Percent Response	Valid: Percent Response	Valid: Cumulative Percent Response	Train: Percent Response	Train: Cumulative Percent Response	Valid: Percent Response	Valid: Cumulative Percent Response
1.84	1.81	93.9	96.9	93.9	92.4	9.71	20.0	9.74	19.1
1.75	1.85	98.8	98.7	97.8	89.3	10.3	20.6	8.78	18.2
1.59	1.82	100	100	84.7	80.0	7.99	15.9	7.99	15.9
1.59	1.59	77.2	77.2	77.0	77.0	7.99	15.9	7.99	15.9

Diagram 9.6 Fit Statistics Table 3

Based on the Fit Statistics Table 1, the Regression model yielded the lowest misclassification rate at 0.249% on the validation set. This indicates that the

Regression model has a higher predictive accuracy. On the other hand, both Maximal Tree and Decision Tree models exhibited the highest misclassification rates at 0.264. Hence, these models might be less effective in predicting accuracy.

Besides, the Akaike's Information Criterion (AIC) of the Regression model reported a value of 770.94. This value suggests that the Regression model is relatively efficient. It captures the underlying patterns of the dataset without being overly complex.

According to the Fit Statistics Table 2, the perfect Gini coefficient of 1 for the Random Forest model on the validation set indicates that it has excellent discrimination capacity, perfectly distinguishing between the different outcomes in this particular dataset. On the contrary, the Gini coefficients for the Maximal Tree and Decision Tree models at 0.686 reflect a moderate ability to differentiate between outcomes, suggesting that these models may struggle with more complex patterns within the data.

In addition, the Regression model reported the highest ROC index of 0.909, indicating that it can effectively discriminate between various classes within the data. In contrast, both the Decision Tree and Maximal Tree models exhibited the lowest ROC index values at 0.84. This means that it is weak at distinguishing between classes.

In short, the analysis of the Fit Statistics indicates that the Regression model performs the best. This is evidenced by the lowest misclassification rate at 0.249 and the highest ROC index of 0.909. These findings suggest that the Regression model is well-suited for accurate predictions and effective class separation.

Event Classification Table  
Model Selection based on Valid: Misclassification Rate (\_VMISC\_)

Model	Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
Reg		Regression	TRAIN	Target	Target	24	256	83	295
Reg		Regression	VALIDATE	Target	Target	29	262	79	289
Tree		Maximal Tree	TRAIN	Target	Target	19	238	101	300
Tree		Maximal Tree	VALIDATE	Target	Target	25	239	102	293
Tree2		Decision Tree	TRAIN	Target	Target	19	238	101	300
Tree2		Decision Tree	VALIDATE	Target	Target	25	239	102	293
HPDMForest		Random Forest	TRAIN	Target	Target	.	339	.	319
HPDMForest		Random Forest	VALIDATE	Target	Target	25	245	96	293
Boost		Gradient Boosting	TRAIN	Target	Target	16	249	90	303
Boost		Gradient Boosting	VALIDATE	Target	Target	30	255	86	288

**Diagram 9.7 Event Classification Table**

Based on Diagram 9.7, the Regression model has the consistent performance between training and validation datasets. In the training phase, the model has 24 false negatives and 83 false positives, with a strong presence of true positives (295) and true negatives (256). In the validation phase, there's a slight increase in false negatives to 29 and a decrease in true positives to 289, with false positives reducing to 79 and true negatives increasing to 262.

## 10.0 Conclusion

The primary aim of this research was to delve into the factors contributing to academic dropouts in higher education and to optimize curriculum performance based on these insights. Leveraging a comprehensive dataset from various higher education institutions, the study employed a range of data mining techniques to predict student outcomes and unravel the key factors influencing academic success.

Through meticulous data preprocessing, including handling missing values, encoding categorical variables, and standardizing data, we ensured the dataset's quality and consistency, setting a strong foundation for accurate model predictions. The exploratory data analysis (EDA) phase was particularly revealing. It highlighted significant insights, such as the impact of debt on student outcomes, the correlation between age at enrollment and semester grades, and the variation of academic outcomes across different courses. These findings pointed towards socioeconomic factors, age, and course selection as significant influencers of student success.

The study then ventured into predictive modelling, employing various models like Decision Tree, Regression, Gradient Boosting, and Random Forest. Each model shed light on different aspects of the factors affecting student outcomes, with the Regression model standing out for its high accuracy and effectiveness in predicting student outcomes. This was further corroborated in the model comparison and evaluation phase, where the Regression model emerged as the most effective, boasting the lowest misclassification rate and the highest ROC index, indicating its superior ability in accurately predicting and differentiating between various student outcomes.

The implications of these findings are far-reaching. For one, they underscore the need for curriculum optimization based on identified factors such as debt status, age at enrollment, and course selection. Institutions can tailor their curriculum and

support strategies, particularly by providing financial advice and support to reduce dropout rates among students grappling with debt. Furthermore, the early identification of at-risk students using predictive models paves the way for timely interventions, allowing for the development of tailored support programs to cater to specific needs, such as additional tutoring for older students or more resources for courses with higher dropout rates.

All in all, this research underscores the transformative potential of data mining techniques in understanding and predicting student outcomes in higher education. By tapping into the power of data analytics and machine learning, educational institutions can gain invaluable insights into student performance and dropout dynamics, enabling them to craft more informed and effective educational policies and practices. This not only enhances the educational experience for students but also contributes significantly to the broader goal of improving educational outcomes and fostering a more supportive and effective learning environment.

## 11.0 References

*Basic Plots and Charts.* (n.d.). SAS Help Center.

[https://documentation.sas.com/doc/en/pgmsascdc/9.4\\_3.5/grstatproc/n19gxtzyuf79t3n16g5v26b73ckv.htm#n0iuhw0kbsunrqn1lep9q1c8mrv0](https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/grstatproc/n19gxtzyuf79t3n16g5v26b73ckv.htm#n0iuhw0kbsunrqn1lep9q1c8mrv0)

Ghose, A. M. A. (2011). Decision Tree Induction & Clustering Techniques In SAS Enterprise Miner, SPSS Clementine, And IBM Intelligent Miner A Comparative Analysis. *International Journal of Management & Information Systems*, 14(3).

<https://doi.org/10.19030/ijmis.v14i3.841>

*Module 2 - Process.* (n.d.). learn.theprogrammingfoundation.org.

[https://learn.theprogrammingfoundation.org/getting\\_started/intro\\_data\\_science/module2/?gclid=Cj0KCQiA1rSsBhDHARIsANB4EJZ96rNQNwKQhaoK3WMXv1BdrjlaRwHk1TmevBNDBQBJkNQsWjr180aAtNSEALw\\_wcB](https://learn.theprogrammingfoundation.org/getting_started/intro_data_science/module2/?gclid=Cj0KCQiA1rSsBhDHARIsANB4EJZ96rNQNwKQhaoK3WMXv1BdrjlaRwHk1TmevBNDBQBJkNQsWjr180aAtNSEALw_wcB)

Rithika, S. (2023, January 18). *Sequence data in Data Mining Simplified 101 - learn.* Hevo. <https://hevo.com/learn/sequence-data-in-data-mining/>

SAS help center. (n.d.).

[https://documentation.sas.com/doc/da/vdmmlcdc/v\\_017/vdmmlref/n0cly6g8j7ot3ln1tkouhoizzgh8.htm](https://documentation.sas.com/doc/da/vdmmlcdc/v_017/vdmmlref/n0cly6g8j7ot3ln1tkouhoizzgh8.htm)

Vanawat, N. (2023, May 5). How To Perform Exploratory Data Analysis -A Guide for Beginners. Analytics Vidhya.

<https://www.analyticsvidhya.com/blog/2021/08/how-to-perform-exploratory-data-analysis-a-guide-for-beginners/#h-how-to-perform-eda>

*What is exploratory data analysis?* (n.d.). IBM.

<https://www.ibm.com/topics/exploratory-data-analysis>