**DATA MINING INSIGHTS TO DISCOVERING PATTERNS IN STUDENT ACADEMIC PERFORMANCE**

**BY**

**ANI,** CHIDERA PRISCILLA

(17CG023150)

**A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES, COLLEGE OF SCIENCE AND TECHNOLOGY, COVENANT UNIVERSITY OTA, OGUN STATE.**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE BACHELOR OF SCIENCE (HONOURS) DEGREE IN COMPUTER SCIENCE**

**JULY 202****1**

1. CERTIFICATION

I hereby certify that this project was carried out by Chidera, Priscilla ANI in the Department of Computer and Information Sciences, College of Science and Technology, Covenant University, Ogun State, Nigeria, under my supervision.

1. Name: Dr. Olufunke O. Oladipupo

(**Supervisor)**

Signature ­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date ­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Name: Dr. Olufunke O. Oladipupo

(**Head of Department)**

Signature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. DEDICATION

This report is dedicated to God Almighty for his unmerited love and grace in my life. He constantly shows up for me, and I am eternally grateful.

I also dedicate it to my parents and my brother, who show me immense support, love, and understanding all the time.

1. ACKNOWLEDGEMENTS

Special acknowledgement goes to God for His guidance, grace, favour, and protection throughout my life and for making this project possible even when I was in doubt. Indeed, all things are possible through him.

I acknowledge all my family members and friends who have shown me love and support throughout the period I worked on this project.

Also, my supervisor, Dr Olufunke Oladipupo, was there every step of the way, helping me bring this project to life and for every word of advice and encouragement. Thank you, ma.

To all Computer and Information Science staff members who have imparted me with the knowledge to make this project a reality, especially Dr Afolabi, who was constantly checked on me and was ready to help in any way.

Finally, to all my course mates. These four years have been a lot, and I enjoyed every moment of it. I am incredibly grateful to my friends and people that helped me in any way to complete this project.

# TABLE OF CONTENTS

**Title Page**

[Cover page](#_Toc79579040)

[Title page i](#_Toc79579040)

[Certification ii](#_Toc79579040)

[Dedication iii](#_Toc79579041)

[Acknowledgements iv](#_Toc79579042)

[Table of contents v](#_Toc79579043)

[List of figures viii](#_Toc79579044)

[List of tables x](#_Toc79579045)

[Abstract xi](#_Toc79579046)

**CHAPTER ONE: INTRODUCTION**

[1.1 Background Information 1](#_Toc79579049)

[1.2 Statement of the Problem 3](#_Toc79579050)

[1.3 Aim and Objectives of the Study 3](#_Toc79579051)

[1.4 Methodology 4](#_Toc79579052)

[1.5 Significance of the Study 4](#_Toc79579053)

[1.6 Limitations of the Study 5](#_Toc79579054)

[1.7 Project Organisation 5](#_Toc79579055)

**CHAPTER TWO: LITERATURE REVIEW**

[2.1 Introduction 6](#_Toc79579063)

[2.2 Data Mining 6](#_Toc79579064)

[2.3 Data Mining Techniques 9](#_Toc79579065)

[2.4 Education Data Mining 11](#_Toc79579066)

[2.5 Descriptive Analytics 13](#_Toc79579067)

[2.6 Student Academic Performance 14](#_Toc79579068)

[2.7 Student Performance Analysis 15](#_Toc79579069)

[2.8 Summary 25](#_Toc79579070)

**CHAPTER THREE: RESEARCH DESIGN**

[3.1 Introduction 27](#_Toc79579076)

[3.2 Requirement Analysis 27](#_Toc79579077)

[3.2.1 Functional Requirement 27](#_Toc79579086)

[3.2.2 Non-Functional Requirement 28](#_Toc79579087)

[3.3 System Architecture 28](#_Toc79579088)

[3.4 System Design 29](#_Toc79579089)

[3.4.1 Physical Design 29](#_Toc79579094)

[3.4.2 Logical Design 30](#_Toc79579095)

[3.4.3 Data Pre-Processing 34](#_Toc79579096)

[3.5 Research Methodology 32](#_Toc79579097)

[3.5.1 Data Collection 33](#_Toc79579100)

[3.5.2 Data Analysis 34](#_Toc79579101)

[3.5.3 Result Interpretation/ Inferencing 43](#_Toc79579102)

[3.6 Conclusion 43](#_Toc79579103)

**CHAPTER FOUR: SYSTEM IMPLEMENTATION**

[4.1 Introduction 44](#_Toc79579107)

[4.2 System Requirements 44](#_Toc79579108)

[4.2.1 Minimum Hardware Requirements 44](#_Toc79579118)

[4.2.2 Minimum Software Requirements 44](#_Toc79579119)

[4.3 Analysis Result 45](#_Toc79579120)

[4.3.1 Descriptive Analysis Result 45](#_Toc79579123)

[4.3.2 Data Visualisation Result 49](#_Toc79579124)

[4.3.3 Associative Analysis Result 53](#_Toc79579125)

[4.3.4 Clustering Result 56](#_Toc79579126)

[4.4 Result Interpretation 63](#_Toc79579127)

[4.5 Program Modules and Interfaces 63](#_Toc79579128)

[4.5.1 Home Page 63](#_Toc79579140)

[4.5.2 Registration Page 64](#_Toc79579144)

[4.5.3 Login Page 64](#_Toc79579145)

[4.5.4 View Association Page 65](#_Toc79579146)

[4.5.5 Course Association Mining 66](#_Toc79579147)

[4.5.6 Course Performance Prediction Page 66](#_Toc79579148)

[4.5.7 Dashboard 67](#_Toc79579149)

**CHAPTER FIVE: SUMMARY, RECOMMENDATION, AND CONCLUSION**

[5.1 Summary 69](#_Toc79579153)

[5.2 Recommendation 69](#_Toc79579154)

[5.3 Conclusion 70](#_Toc79579155)

[References 71](#_Toc79579156)

[Appendix 77](#_Toc79579157)

1. LIST OF FIGURES

**FIGURE PAGE**

[Figure 2.1 KDD Process (Kamber et al., 2012) 7](#_Toc79650716)

[Figure 3.1 Architecture of the system (Hujer, 2011) 29](#_Toc79650717)

[Figure 3.2 Use Case diagram for the system 31](#_Toc79650718)

[Figure 3.3 Activity diagram for the system 32](#_Toc79650719)

[Figure 3.4 Research Methodology Workflow 33](#_Toc79650720)

[Figure 3.5 Sample questionnaire 34](#_Toc79650721)

[Figure 3.6 Diagrammatic representation of Apriori algorithm 39](#_Toc79650722)

[Figure 3.7 Example of elbow point 42](#_Toc79650723)

[Figure 4.1 Importing libraries and loading dataset 46](#_Toc79650724)

[Figure 4.2 Replacing missing values 46](#_Toc79650725)

[Figure 4.3 Descriptive analytics of dataset 47](#_Toc79650726)

[Figure 4.4 Distribution of variables 47](#_Toc79650727)

[Figure 4.5 Histplot of CGPA 48](#_Toc79650728)

[Figure 4.6 Distribution of CGPA 48](#_Toc79650729)

[Figure 4.7 Correlation between variables 49](#_Toc79650730)

[Figure 4.8 Frequency of male and female students 50](#_Toc79650731)

[Figure 4.9 Distribution of male and female students by CGPA 50](#_Toc79650732)

[Figure 4.10 Average GPA performance by gender 51](#_Toc79650733)

[Figure 4.11 Class distribution of students 51](#_Toc79650734)

[Figure 4.12 Current vs Previous CGPA 52](#_Toc79650735)

[Figure 4.13 Performance distribution of all courses 52](#_Toc79650736)

[Figure 4.14 Loading in the dataset 53](#_Toc79650737)

[Figure 4.15 One-hot encoding using TransactionEncoder() 54](#_Toc79650738)

[Figure 4.16 Generating the itemset 54](#_Toc79650739)

[Figure 4.17 The generated rules 55](#_Toc79650740)

[Figure 4.18 Loading the dataset 55](#_Toc79650741)

[Figure 4.19 Generating itemset and rules 56](#_Toc79650742)

[Figure 4.20 Resulting rules 56](#_Toc79650743)

[Figure 4.21 Clustering dataframe 57](#_Toc79650744)

[Figure 4.22 Scaling variables 57](#_Toc79650745)

[Figure 4.23 Implementing the elbow method 58](#_Toc79650746)

[Figure 4.24 Result of using the elbow method 58](#_Toc79650747)

[Figure 4.25 Implementing silhouette coefficient 59](#_Toc79650748)

[Figure 4.26 Result of using silhouette coefficient 59](#_Toc79650749)

[Figure 4.27 Implementing gap statistics 60](#_Toc79650750)

[Figure 4.28 Result of using gap statistics 60](#_Toc79650751)

[Figure 4.29 Clustering the variables 61](#_Toc79650752)

[Figure 4.30 Number of students in each cluster 61](#_Toc79650753)

[Figure 4.31 Adding cluster labels to the dataframe 62](#_Toc79650754)

[Figure 4.32 Profiling chart from clustering 62](#_Toc79650755)

[Figure 4.33 Home Page 63](#_Toc79650756)

[Figure 4.34 Registration Page 64](#_Toc79650757)

[Figure 4.35 Login Page 65](#_Toc79650758)

[Figure 4.36 View Association Page 65](#_Toc79650758)

[Figure 4.37 Course Association Mining 66](#_Toc79650758)

[Figure 4.38 Course Prediction 67](#_Toc79650758)

[Figure 4.39 Dashboard Page 67](#_Toc79650758)

[Figure 4.40 Loading model into Django web app 68](#_Toc79650758)

[Figure 4.41 Generating charts with chartjs 68](#_Toc79650758)

1. LIST OF TABLES

**TABLE PAGE**

[Table 4.1 Hardware Requirements 44](#_Toc79650698)

[Table 4.2 Software Requirements 45](#_Toc79650699)

1. ABSTRACT

The educational sector plays a very vital role in the life of an individual. A person's ability to develop and hone abilities that may be utilised in various areas can be enhanced via education. Because of its significance, academics and educational institutions are constantly incorporating data mining techniques to help students succeed. This study attempts to improve this sector by examining academic performance patterns and investigating factors that may influence student achievement.

To achieve the aim of this project, the effect of gender on students’ performance was visualised using visual plots. Students were segmented based on their raw scores in different subjects using the K-Means clustering algorithm. The associative Apriori algorithm was modelled on the dataset and interesting associations were generated. A system that enables easy interaction with the model was also built to be used by administrators and educational institutions.

From the rules, generated it is observed that a moderate curriculum, a practical course, fair student capacity, and a moderate course structure are the factors that lead to a good course overall performance. The student pattern mining system aims to serve as an interactive system that should be used for educational administrative purposes and student self-approval to enhance student performance.

# CHAPTER ONE

# INTRODUCTION

## BACKGROUND INFORMATION

As information technology continues to evolve globally, there has been a surge in data collected and released daily. This surge resulted in the challenge of storing and analysing these data. Such developments have led to terms such as 'big data,' 'data warehousing,' 'machine learning,' and 'data mining.' Data Mining is the process of analysing vast amounts of data sets to discover patterns and make meaningful decisions (Şen et al., 2012). It offers sophisticated automated tools that can be used to discover previously unknown valid patterns and relationships. Data mining has been previously applied in various domains like telecommunication, retail, healthcare, customer relationships, fraud detection, financial banking, etc.

Educational institutions have begun to leverage the vast amount of data available through various analyses to improve their performance. They analyse prospective students' records during the admission process by going through their past academic achievements in their previous educational institution to determine the best students to recruit. This has become necessary for several institutions because a high-quality university is based on an excellent academic record (Shahiri et al., 2015). Thus, such institutions need to recruit students with high grades and ensure that necessary procedures are implemented to achieve even better results. This is where students' performance prediction comes into play.

Over recent years, researchers have grown curious about determining the elements that affect the performance of students. Researchers like Mohamad and Usman (2013) say student performance is obtainable by measuring the learning assessment and curriculum or measured by graduation (Rastrollo-Guerrero et al., 2020). In Nigerian higher institutions, it is based on a students' ability to ace graded tests and exams and maintain an excellent Cumulative Grade Point Average (CGPA). This is tricky as a student's performance should not be solely restricted to grading methods and scores. This has led to a need to apply data mining techniques to the academic sector to determine factors affecting students' learning outcomes.

The academic performance of a student is a measure of the extent to which the student has performed. With the low-performance rate of higher institution graduates and a rise in school dropout rates in the country, there is an ascent in need to investigate the degree of retention, proper distribution of education resources, and mediation procedures that strongly influence students' academic performance. This can be determined in various ways, such as curriculum assessment, graduation rates, extra-curricular activities, and assessment scores. Analysing students' performance can help make predictions that would help restructure the learning process to benefit the institution. It could assist in proper course selections, identify struggling students, identify factors that affect the students' learning, measure lecturers' performance, and admit the right students. These predictions will be to formulate strategies that would improve academic achievements based on said findings.

EDM (Educational Data Mining) is a new area derived from data mining. It is concerned with the development, study, and use of computerized systems for detecting patterns in huge collections of educational data that would otherwise be impossible to analyze owing to the vast volume of data (Romero et al., 2010). The goals of EDM include predicting students' future learning behaviour, identify students' learning patterns, contemplate the effects of academic support, and advance the knowledge base about learning. A student's academic achievement is influenced by a variety of factors, including personal, socioeconomic, psychological, and other environmental characteristics (Kaur et al., 2015).

Due to the increasing dropout rate, and the need for higher institutions to avoid financial loss, it has become necessary to understand the underlying factors and patterns that could hinder students' performance. Such patterns could differ across academic institutions based on the elements at play in the educational environment, such as the curriculum, management, and other social factors. Due to this, educational establishments must weigh in the various factors at play in their environment and build effective predictive models that capture their students and the institution at large. Higher institutions have recently begun to take advantage of EDM's possibilities to increase student achievement and engagement, thereby elevating their institution's credibility.

## STATEMENT OF THE PROBLEM

Educational Data Mining considers the analysis of education datasets to discover patterns in the data, which will help educators provide a practical teaching approach, monitor their students' achievements, and answer related research questions.

Although EDM has been in existence for a while, many institutions, especially in Nigeria, are yet to leverage its possibilities. A common challenge institutions face is deciding how to capture, organise, understand, learn, and productively use this data acquired in the recruitment, retention, and planning of academic activities (Romero et al., 2010).

Various researches have predicted students' performance in terms of their future GPA and dropout possibility (Sandoval et al., 2018). Still, it is imperative to consider some other key factors affecting students' performance and propose strategies that the institution could take to improve such circumstances. This research aspires to apply some previous research techniques and dives deeper into analysing some critical underlying factors that could affect student performance.

## AIM AND OBJECTIVES OF THE STUDY

The study aims to explore data mining techniques to discover interesting patterns in students' academic performance with an interactive system.

The objectives necessary to achieve this aim include:

1. To gather and store all the required datasets needed for the analysis.
2. To pre-process the dataset and carry out descriptive analytics on the dataset.
3. To investigate the effect of gender on student performance.
4. To examine the predictive impact of student capacity, course structure, course nature and course curriculum broadness on overall performance.
5. To establish an associative relationship between students' failed courses and between students' pass grades.
6. To group students based on their performance.
7. To develop an interactive interface for the data analytic process.

## METHODOLOGY

This research study would follow a defined methodology which includes:

1. Academic and demographic data of students was obtained from the Department of Computer and Information Science, Covenant University. Expert opinion on course details from the department's lecturers and final year students.
2. Visualisations approach with histogram, bar chart, distribution plot, and box plot were used for descriptive analytics. Python programming will be used for necessary implementation.
3. Visual plots such as bar charts and box plots would be used to examine the effect of gender on performance, and this task was implemented using Python programming.
4. An associative algorithm, the Apriori model, was used to investigate the predictive impact on-course performance.
5. Use of descriptive analytics and an associative algorithm (Apriori algorithm) to determine the relationship between students who failed and passed courses and their grades.
6. Use of K-Means clustering algorithm to cluster students using their raw scores.
7. The interface was designed using HTML, CSS, JavaScript, and Django.

## SIGNIFICANCE OF THE STUDY

The study analyses student result data and conduct a thorough analysis of attributes that can potentially impact students' academic outcomes in the university by taking the course nature and students' gender into consideration.

It would provide an in-depth insight that would enable management to make effective decisions to assist a student in improving or maintaining their academic performance. By this means, proper counselling and enlightening seminars can be hosted to guide students and drive them closer to their educational goals.

Results obtained from the study would also provide a practical approach to the admission process of an institution by helping to admit the right students selectively. This research also takes a dynamic approach by grouping and predicting students' performance based on their raw scores obtained rather than using a more general approach – CGPA.

Finally, this research would also educate on some machine learning algorithms and their effectiveness in predicting student performance and determining the existing relationships in student learning.

## LIMITATIONS OF THE STUDY

Many factors can influence student academic performance. This work focuses on using datasets of student biodata, grade scores, and course nature.

The study will make use of data obtained from the Department of Computer and Information Science in Covenant University. This study does not consider other university departments and does not consider other socio-economic factors affecting student performance.

## PROJECT ORGANISATION

Chapter One presents an overview of the project, the background information, problem statement, research objectives, significance, scope, and limitation of the study. Chapter Two discusses an overview of the literature on existing systems and principles. Chapter Three describes the methodology in detail, the research design, data description, and analysis techniques. Chapter Four presents the results generated from the analysis. Chapter Five summarises the project, gives recommendations, suggestions, and conclusions.

# CHAPTER TWO

# LITERATURE REVIEW



## INTRODUCTION

In this modern age, educational data has become a valuable resource that contributes significantly to society's overall well-being. To stay afloat, educational institutions must use EDM to improve the quality of education in their institution. To do this, various studies have combined different variables, approaches, and algorithms to forecast student academic success.

This chapter gives an overview of concepts related to data mining and education data mining. It reviews existing works of literature of other researchers, focusing on their aim, data used, methodology, findings, limitations, and areas for future study. Finally, it summarises results from the reviewed works of literature and proposes a new directive.

## DATA MINING

The evolution of information technology has led to a vast increase in data (structured and unstructured) made available every day (Baradwaj & Saurabh, 2011). There has become an urgent need to analyse useful information and extract meaningful patterns from the data. The term 'data mining' is a misnomer because it fails to accurately capture the purpose of the word, which should be to mine knowledge from data (Kamber et al., 2012). Thus, data mining's primary goal is to extract information from a data set and transform it into an understandable format for further use and predictions.

Data mining is the process of analysing large amounts of data to discover trends, insights and predict an outcome or target. It is an interdisciplinary field that uses statistics, machine learning, and Artificial Intelligence (AI) to analyse data and forecast future events. Data mining turns an extensive collection of data into knowledge. It is often used interchangeably with the term Knowledge Discovery in Data (KDD) since it is a core part of the knowledge discovery process (Kamber et al., 2012). Figure 2.1 shows the KDD process.

Data mining has been applied in various domains, including retailing, banking, medical, insurance, education, customer relationship management, transportation, telecommunications, and criminal investigation.

Diagram

Description automatically generated

Figure 2.1 KDD Process (Kamber et al., 2012)

In the banking domain, Aburrous et al. (2010) proposes a methodology for identifying e-banking phishing websites that is intelligent, robust, and competitive. Data mining algorithms alongside fuzzy logic were combined to identify the factors and strategies that constitute the phishing website. A final risk rating of 72% accuracy was obtained. Liébana-cabanillas et al. (2013) examined various techniques for assessing variables that are vital for predicting possible levels of confidence among e-banking users. The best method for variable selection was obtained using Mutual Information alongside K-NN. Nagpal and Mishra (2021) evaluated the need for and importance of HR analytics for making decisions in the banking sector. Questionnaires were distributed to bank managers of private and public banks, and analytics results indicated that most banks preferred HR analytics for smarter business judgments.

Various approaches to classifying procedures, forecasting patient categories, and administering the best remedy have significantly improved the healthcare field. Lee et al. (2020) applied data mining to explore the relationships between medicinal herbs and skin-related keywords from the Donguibogam text applied to 626 medicinal herbs. They were able to provide a detailed description of 52 medicinal herbs for skincare. Ramani and Sivagami (2011) conducted a survey with the aid of data mining algorithms to discover techniques for classifying Parkinson's disease. Feature relevance analysis was carried out, and various classification algorithms were modelled on the data set. The Random Tree classifier emerged as the best model. Chen et al. (2018) used logistic regression to perform a case study to determine the associations between dementia and varying illnesses. Patients were divided into case patients (those prescribed specific dementia medications) and control patients (those who had no history of dementia diagnosis or medication orders for dementia drugs). The data was analysed using a logistic regression model. The result of the analysis identified other illnesses that were associated with dementia patients of a specific demographic.

Customer relationship management, according to Swift (2001), is an "enterprise approach to recognising and manipulating customer behaviour through positive interactions to increase customer acquisition, retention, loyalty, and profitability." Wei et al. (2013) utilised data mining techniques to segment customers and develop marketing strategies for a hair salon by integrating self-organising maps (SOM) as well as K-means clustering in an RFM (recency, frequency, and monetary) model. They were able to formulate unique marketing strategies for the customers using the clustering model, which identified four different customer types. Jiang et al. (2019) introduces a DENFIS (dynamic evolving neural-fuzzy inference system) alongside opinion mining to dynamically model customer preferences using time series data based on online customer reviews. Yoon et al. (2020) devised a method for evaluating customer feedback and applying it to technology creation strategy to improve a product's perceived value among customers. The study recommends using the Structural Equation Model analysis to determine the automotive industry's technical level and consumer satisfaction, considering the moving system's characteristics.

Data mining has also been applied extensively in the criminal investigation domain. Fraud is a significant problem for merchants, particularly in the online sector. Carneiro et al. (2017) developed and implemented a fraud detection system in a major e-tail merchant. Ngai et al. (2011) reviewed numerous literature on financial fraud detection and listed four financial fraud techniques and various data mining techniques for fraud detection. Varela et al. (2020) used data mining techniques to analyse malicious messages in banners, media platforms, and other databases to propose a geospatial model to classify and systematically distribute the message's authors. They were able to classify suspected criminals by analysing these messages' content using natural language processing (NLP) and artificial intelligence (AI) techniques.

## DATA MINING TECHNIQUES

Data mining incorporates techniques from various disciplines as a multidisciplinary field, including statistics, machine learning, database management systems, neural networks, rough sets, and visualisation (Fu, 1997). Data mining tasks are divided into two groups based on the task at hand, which are descriptive and predictive tasks. Descriptive data mining aims to uncover trends in the data, while predictive data mining uses the existing data set to infer how a new data set would behave. Some of the data mining techniques include:

1. **Classification:** this is a machine learning technique that separates the objects or variables in a data collection into pre-defined groups or classes. Its goal is to figure out what class an unlabelled data case belongs to. Linear programming, statistics, decision trees, and artificial neural networks are all used in data mining. Classification is a modelling approach for developing applications that can categorize items in a data collection into different categories. Classification algorithms include Logistic Regression, Naïve Bayes, K-Nearest Neighbour, Decision Trees, Random Forest, Support Vector Machine.
2. **Clustering:** this is an unsupervised machine learning algorithm that finds trends in the unlabelled data by grouping identical data points to the same cluster and different data points to other clusters. Unlike classification that deals with pre-defined classes, clustering defines the class of a data point. Clustering uses visual methods to make data understandable. The objective is to maximise intraclass similarity while reducing interclass similarity. When common groupings in the data are unknown, the clustering approach is utilized, and it may also be used to minimize the size of the research region. There are two forms of cluster analysis: hierarchical clustering and non-hierarchical clustering. Single linkage, full linkage, average linkage, median, and Ward are hierarchical clustering approaches, whereas k-means, adaptive k-means, k-medoids, and fuzzy clustering are non-hierarchical clustering techniques (Abaidullah et al., 2015). Some clustering algorithms include K-Means, Mean-Shift, Density-Based, Expectation-Maximization.
3. **Prediction:** This method forecasts missing or potential data by looking for trends in historical or current data values. It establishes a connection between one or more independent (previously determined) variables and dependent (yet to be chosen) variables. Predictions are forecasts of what will occur in the future. For the output variable, the prediction model must contain limited labelled data. The labelled data provides some insight into the factors that need to be predicted. Prediction analysis methods include machine learning (regression, classification), artificial intelligence (neural networks), and descriptive analytics. Regression analysis is a data mining technique for calculating numerical estimations. Regression methods include linear regression, nonlinear regression, multivariate regression, and multivariate nonlinear regression.
4. **Association Rule:** The term "association" refers to a data mining methodology related to statistics. It denotes a pattern based on a specific item's relationship to other things in the same transaction and identifies trends in events simultaneously. It is analogous to the machine learning principle of co-occurrence, in which the existence of one data-driven event suggests the probability. Relationship mining aims to see whether one event triggers another by looking at the overlap between the two events in the data, which can be done using if-then rules. Correlation is a mathematical term that is close to the concept of association. Market basket research, customer relationship management, medical diagnosis, and census data may benefit from association rules. Some association algorithms are Apriori, Eclat, and Frequent-Pattern growth (FP-growth).
5. **Sequential Patterns**: This data mining technique is based on uncovering a set of events that occur in a predetermined, sequential order. It is beneficial when mining transactional data. It can be very helpful in assisting companies in recommending new products to consumers to improve sales. The critical distinction between sequence pattern mining and association rule mining is that the former considers the time sequence. Time-series mining, which deals with time-series data over a specific period, is often closely related to sequential patterns.
6. **Summarisation:** This deals with the generalisation of data. It is a primary data mining technique for finding a concise overview of a dataset. For exploratory data analysis, data visualisation, and automated report production, simple summarisation methods such as tabulating the mean and standard deviations are frequently used (Chandola & Kumar, 2007). Summarisation aims to condense a large dataset into a smaller collection of patterns while preserving as much detail as possible. The four key areas to consider when summarising are replication, centrality, dispersion, and shape.

## EDUCATION DATA MINING

A significant problem faced by educational institutions is the rapid increase of data and the need to use this data to improve academic performance and making managerial decisions (Abaidullah et al., 2015; Delavari et al., 2008). Educational Data Mining (EDM) is an emerging field that applies data mining techniques to educational data (Kaur et al., 2015). It is concerned with creating strategies for investigating information from academic settings to provide formal education to students (Baradwaj & Saurabh, 2011). EDM can be defined as applying traditional data mining techniques to educational data analysis to solve problems in the academic context (Baker & Yacef, 2009).

According to Prabha and Shanavas (2014) the following are defined as the goals of EDM:

1. Develop student models that leverage detailed information such as students' experience, meta-cognition, motivation, and attitudes to predict students' potential learning actions.
2. To develop domain models that describe the material to be learned as well as the best instructional sequences.
3. To investigate the effects of various types of pedagogical support that learning software can provide; and
4. To advance empirical data about learning and learners by developing computational models that combine student, software, and domain models.

There exists a wide range of applications for educational data mining. Some of them include the development of e-learning systems (Cantabella et al., 2018; Castro et al., 2007; Lara et al., 2014), student performance prediction (Fernandes et al., 2019; Kaur et al., 2015; Rastrollo-Guerrero et al., 2020), pedagogical support (Seufert Sabine, Christoph Meier, Matthias Soellner, 2019), predicting slow learners (Kaur et al., 2015; Ramaswami & Bhaskaran, 2010), and educational clustering data (Castro et al., 2007; Delavari et al., 2008).

Romero and Ventura (2010) categorized the main applications that use educational data mining techniques as data analysis and visualisation, feedback for instructors, student recommendations, predicting student performance, student modelling, detecting undesirable student behaviour, grouping students, social network analysis, developing concept maps, planning, scheduling, and courseware construction.

EDM methods can help teachers and students develop their teaching and learning processes in the classroom, recognise at-risk students, adapt teaching strategies, and make recommendations. The EDM techniques that are most widely used include classification (decision trees, logistic regression, Naïve Bayes, support vector machines, etc.), regression analysis, association rules (Apriori algorithm), clustering (k-means clustering, hierarchical-based clustering), neural networks (artificial neural network, convolutional neural network), discovery within models, and dimensionality reduction techniques.

## DESCRIPTIVE ANALYTICS

Descriptive analysis is one of the preliminary steps in data analysis, and it involves using maps, tables, and other suitable visualisations to display or summarise data. Descriptive statistics are a collection of statistical methods used to summarise data gathered during an inquiry (Fernandes et al., 2019). It aids in the generalisation of data distribution, identifying attributes with outliers, and identifying established relationships between variables. Descriptive analysis is an essential step in data mining because it allows users to get a quick overview of the dataset. After the model has been developed, this knowledge can be double-checked. Before constructing their model, numerous researchers have used descriptive analytics to extract trends from their dataset. The following are some examples of literature that have used descriptive analytics in educational data mining.

Cantabella et al. (2018) used visualisation to pinpoint an increase in student participation in a given year due to implementing a new online learning method. Using a descriptive study, Fernandes et al. (2019) was able to visualise a rise in the failure rate from 12.5168% in 2015 to 13.0854% in 2016. They were able to create a model to decrease the rate using this information. Altabrawee et al. (2019) was able to extract information from student data to determine the percentage of weak students who worked or were married. Cerezo et al. (2016) used descriptive analytics to assess the average age and percentage of females in the population before conducting the analysis. Since schooling is optional for males and not intended for females in most parts of Northern Nigeria, Goga et al. (2015) was able to identify the sparse distribution of data samples for students from that region. Popoola et al. (2018) used visualisation charts (boxplots, histograms, comparison plots, pie charts) and Analysis of Variance (ANOVA) to visualise the distribution and skewness of students' GPA and determine if there is a significant variation across all engineering disciplines.

## STUDENT ACADEMIC PERFORMANCE

A significant application of education data mining is in the prediction of student performance. Student performance prediction can be used to predict students' final scores in a course, GPA, drop out/failure rate, success in a course, and at-risk students. Attributes and prediction methods are the two most prominent factors in predicting student performance (Shahiri et al., 2015).

The attributes could also be viewed as metrics through which performance can be evaluated. According to Kaur (2015), student performance prediction entails predicting the unknown value of a variable that defines the student. Researchers have used various metrics to analyse student performance prediction. A prevalent one is students' Grade Point Average (GPA) and grades (Altabrawee et al., 2019; Ibrahim & Rusli, 2007; Quadri & Kalyankar, 2010). This metric is commonly used because most educational institutions in countries like Nigeria base the academic achievement of their students on their cumulative GPA (CGPA). Also, various analyses reveal grades/GPA as one of the most significant variables.

Another variable frequently used is students' demographics (Fernandes et al., 2019; Ibrahim & Rusli, 2007). This includes age, family background, gender, race, marital status, education level, disabilities. The diverse demographics of students usually influence their academic performance. Other variables that have been used include extra-curricular activities students participate in, the high school background of a student, and psychometric factors such as student engagement, study behaviour, and student interest (Cantabella et al., 2018; Caruth, 2018; Hussain et al., 2018).

Predictive modelling is commonly used to predict student performance. Prediction modelling broadly entails classification, regression, and clustering.  Decision Trees (Altabrawee et al., 2019; Fernandes et al., 2019; Şen et al., 2012), Naïve Bayes (Altabrawee et al., 2019; Rastrollo-Guerrero et al., 2020; Shahiri et al., 2015), Support Vector Machines (Şen et al., 2012; Tomasevic et al., 2020), Logistic Regression (Altabrawee et al., 2019; P. H. Chen et al., 2018; Tomasevic et al., 2020), Artificial Neural Networks (Altabrawee et al., 2019; Rastrollo-Guerrero et al., 2020; Şen et al., 2012), K-Nearest Neighbour (Altabrawee et al., 2019; Baradwaj & Saurabh, 2011; Rastrollo-Guerrero et al., 2020; Shahiri et al., 2015), Linear Regression (Ibrahim & Rusli, 2007), and K-Means Clustering (Abaidullah et al., 2015; Delavari et al., 2008; Kaur, 2015) are examples of predictive algorithms. Classification is the most common predictive technique that researchers in this field have used. The Apriori algorithm (Banswal & Madaan, 2016; Cantabella et al., 2018; Kasthuriarachchi & Liyanage, 2019) is an association mining algorithm, which has also been used extensively in educational data mining.

## STUDENT PERFORMANCE ANALYSIS

This section gives a brief review of some existing works of literature related to student academic performance. It highlights the aim, methodology, data, and findings of these works.

Admasu and Teklay (2019) developed a performance prediction model for five high school courses to forecast students' grades for the following semester. The data was collected over three semesters; the first two semesters' scores were used as input variables, and the model estimated the third semester's outcome. To improve accuracy, they divided the labels into smaller spaces and used the Label Powerset (LP) transformation to convert the label sets to a multi-classification task. Three public high schools in Mekelle, Ethiopia, provided data.

In addition to their basic details (gender, age, and scores on the five courses), students were given questionnaires that included questions about their perceptions of education quality, legal guardians, family income, family educational history, tutorial, GPA in grade 10, parents' occupation, and students' perceptions of education. The data was divided into 70% and 30% training and testing sets, respectively. The training set is first broken up into smaller label sets using the Stochastic Block Model (SBM), a randomised partitioning algorithm called Randomised k labELset (RAkEL), and fast-greedy to overcome the LP transformation problem. LP is then used to convert the partitioned data from a multi-label classification to a single-label classification. The data is further fed into the machine learning algorithms; SVM, Random Forest (RF), KNN, and MLP. The majority voting rule is used to estimate the success of the individual labels. The model is evaluated using Hamming loss, Jaccard index, precision, and F1 micro and macro. Concerning each partitioning algorithm, all base-level classifiers were assessed using the evaluation metrics. The SVM classifier performed best overall, mainly when fast-greedy and RAkEL partitioning schemes were used. The LP ensemble model was compared to binary significance and class chains, both types of problem transformation methods. In terms of evaluation metrics, the LP model outperformed them. More training samples with higher label spaces can be used to model a multi-label ensemble model as future work.

Abu Zohair (2019) demonstrated the possibility of using small datasets from mid-sized/start-up universities to build an accurate performance prediction model. The data used originated from the records of 50 recent graduates. Student ID, age, B.Sc. degree, B.Sc. grade, course names, course grades, and teacher names are among the attributes used. The dataset was further divided into two sets, one for all course grades with a 58.1% accuracy baseline and the other for the dissertation grade with a 60.5% accuracy baseline. Microsoft Excel and Python were used for data preparation, and R studio was used to visualise the dataset, normalise, and select variables. The machine learning algorithms demanded that all data types be translated to numeric. The classification algorithms used to train the model were Multiple Perceptron Artificial Neural Network (MLP-ANN), Naïve Bayes (NB), Support Vector Machines (SVM), K-Nearest Neighbour (KNN), and LDA (Linear Discriminant Analysis). The accuracy metric and Cohen Kappa's coefficient are used to evaluate, and the validation method was Leave-One-Out-Cross Validation (LOOCV). The SVM model had the highest accuracy and Kappa coefficient in datasets 1 and 2, with values of (69.7%, 41.7%) and (76.3%, 37.4%), respectively. The LDA model's accuracy and kappa coefficients were 63.2% and 35.1% for dataset 1 and 71.1% and 44.7% for dataset 2. The study also found that identifying key attributes with limited datasets using visualisation and clustering techniques was more effective than using complex classifiers. The study was confined since it relied solely on student administration results. The application of other variables that influence learning outcomes may be a topic for future research.

To anticipate the performance of students in their upcoming examination, Gatsheni and Katambwa (2018) use three machine learning algorithms, SVM, Bayesian Networks, NB, MLP, and decision trees' J48. At the University of Johannesburg, questionnaires were used to collect data on factors that could influence first-year students' results. The available data consisted of 247 records with five input variables (average matriculation performance, self-study, qualified lecturer, lecture attendance, and first-semester average results). WEKA was used to build the predictive models, while accuracy and RMSE measured performance. Naïve Bayes had the highest classifier with an accuracy score of 75.3% and the lowest RMSE of 0.3555. The cost of the model, which is a multiplication of the confusion and loss matrix, was calculated. A confusion matrix is a table that shows the number of correctly and incorrectly predicted instances, while a loss matrix defines penalties for getting the wrong answer. With Bayesian networks, Naïve Bayes, MLP, J48, and SVM, a cost of 181, 163, 165, 180, and 139 were obtained. Hence, the SVM classifier with an accuracy of 72.87% is chosen as the best classifier because of its low cost. This model would help to improve the graduation rate of students in the university by identifying specific factors that affect their performance.

Altabrawee et al. (2019) worked on a research project to create an effective classifier that could predict students' academic performance in a computer science course at Al-Muthanna University's College of Humanities. The impact of internet-based learning and time spent on social media on student success is also explored in this study. The dataset was obtained from the Archaeology and Sociology departments, College of Humanities at Al-Muthanna University from 2015 to 2016, and a survey was filled by students. The dataset contains 161 records (76 male and 85 female) with 20 attributes categorised into personal and lifestyle, studying style, family related, educational environment satisfaction, and student's grades. The predictors are department, gender, studying style, using the internet for study, using extra learning resources, interest in studying computer, computer experience, studying hours, family members education, family help in studying, educational environment satisfaction, has a job, accommodation, residence, married, sports participation, time spent on social media, computer grade course1, English grade course1, and final computer outcome as the target variable. The methodology included feature engineering to construct the students' dataset, data collection, data pre-processing, the development and evaluation of four machine learning models, and the best model to analyse the results. The machine learning models used were a fully connected feed-forward Artificial Neural Network, Naïve Bayes, Decision Tree, and Logistic Regression. Students were labelled weak or good based on their final grades in the computer science subject. Weak students had a final grade of less than 60, while good students scored 60 and above. The dataset contained 75 good and 86 weak students.

Weak students were given a positive value in the target variable since their identification was of more importance. RapidMiner studio was used to train, test, and validate the machine learning models (cross-validation). The ROC index, F-Measure, Precision, and Recall were used to assess the models. Overall, the Artificial Neural Network model was the +-most accurate (77.04), it had the highest ROC index (0.807), and the lowest classification error. The next best performer was the Decision Tree model with the second-highest recall, accuracy, and a ROC index of 0.762. Five key attributes were defined by the decision tree: computer grade-Course1, housing, interest in learning computers, satisfaction with the educational environment, and residency. In this vein, the study found that poor students' performance can be improved by arranging extra sessions and lab work, offering better working conditions, and making topics more interesting.

Kaur et al. (2015) studied high school students' performance to identify slow learners and determine the best classifier for predicting performance. This research used the records of 152 students from a high school as a case study. The data were manually filtered, then transformed to a standard format, and used for variable selection using the Waikato Environment for Knowledge Analysis (WEKA) tool. Variables identified: sex, the institution at a high level, type of board, the medium of instruction, type of school, private tuition, area at the school level, students with mobile, students with computers at home, students having internet access, students' roll number, the internal grade of the student, attendance count, and whether qualified or not (target variable). The classifier used include Multilayer Perception (MLP), Naïve Bayes, Sequential Minimal Optimization (SMO), J48 (a decision tree algorithm), and Reduced Error Pruning Decision Tree (REPTree), and they were all implemented with the WEKA tool. Chi-Squared, Information Gain, Symmetrical Uncertainty, and Relief attributes are used to evaluate the classifiers. Multilayer Perception had the highest accuracy of 75% after implementing and evaluating the model, followed by J48 with 69.73%, SMO with 68.42%, REPTree with 67.76%, and Naïve Bayes with 65.13%. As opposed to other classifiers, MLP ranked high with an F-Measure of 82%. Other researchers can also use this model in different fields such as sports, medicine, and the stock market. Learning and retention capabilities may be added to the model to boost their accuracy.

Abu-Oda and El-Halees (2015) also conducted research using data mining techniques to predict student success and dropout risk. Data was gathered from the ALAQSA university database, which contains 1290 records of computer students from 2005 to 2011. According to the visualisations, more than half of the students in the data set dropped out of their original major (computer science). The label variable is a categorical variable that specifies whether the student would graduate with a computer science major. Other variables for prediction are GPA, sex, average grade before university, region, and scores in the following courses: introduction to computer science, database, programming, algorithm analysis, logic design, and data structures I and II. They substituted missing values, used SMOTE to deal with data imbalances, and discretised values into groups to prepare the data. Converting absolute column values into bins or ranges is the process of discretising a value. This was done for the subject scores, average characteristics, and GPA. To accomplish this, upper limits of 100, 80, 70, 60, and 50 were substituted for the A-F range. The data is then divided into 60% and 30% training and testing sets, respectively. The model is developed using a decision tree, Naïve Bayes, and FP-growth for association mining. 10-fold cross-validation is used to examine all models. The decision tree had 98.14%, Naïve Bayes had a 96%, and the FP-growth algorithm revealed patterns with a 95% accuracy. According to the findings, mastering digital design and algorithm analysis courses significantly impacts predicting student persistence in their major and lowering the risk of dropping out.

The research of Goga et al. (2015) focused on using background factors to construct an intelligent recommender framework that could predict a student's first-year academic achievement. They hoped that management would make better decisions about early intervention strategies for low-performing students using this as a guideline. Data was obtained from Babcock University, Nigeria, from 2001 to 2010. Sixty-eight% of the total population was selected at random. The variables chosen for modelling included: age, gender, parent's marital status, parent's educational level, parent's occupation, SSCE (Secondary School Certificate Examination Grade) score, UME (University Matriculation Examination) score, CGPA (Cumulative Grade Point Average) from the first year. Four respondents were given questionnaires, and they all agreed that family backgrounds play a significant role in a student's first-year success. It is stressed that a student's productivity is enabled by a harmonious home, while the reverse is associated with a disturbed mind and inevitably results in poor performance. The model is built using ten WEKA classification algorithms and multilayer perception. The ten classification algorithms are divided into two categories: induction algorithms, including PART, OneR, Decision table, and JRip, and decision tree algorithms, which include REPTree, J48, Random Tree, Decision stump, and Random Forest. The accuracy score was used to assess the classification models. The inductive algorithms performed worse than the decision tree algorithms (REPTree, J48, Random Tree, Decision stump, and Random Forest), where the lowest accuracy was 96.78%. With an accuracy score of 99.908% for 10-fold cross-validation and 99.8205f% for the holdout process, the Random Forest classifier surpassed all other classifiers. The random tree was also used to construct a recommender system that could forecast first-year students' academic success and make suggestions for improved performance. A research extension will use a broader dataset to find rules that correctly simulate real-life scenarios and add more variables from college entry to boost the model's accuracy. The model created could be extended to other academic levels as well.

Ünal (2020) suggests three data mining techniques for predicting students' grades in Math and Portuguese classes. Decision trees, random forests, and Naïve Bayes are the algorithms proposed. Academic records and questionnaires provided to Portuguese high school students were used to collect data. School, age, sex, parents' status, mother and father's education status and occupation, nature of family relationships, family size, guardian, the reason for choosing the school, travel time from home to school, study time per week, amount of class failures, extra-curricular activities, extra paid classes, extra educational support, family educational support, internet access at home, nursery school attended, if the student wants a higher education, romantic relationship, free time after school, going out with friends, alcohol consumption on weekends and workdays, number of school absences, the grade for the first period, the grade for the second period, and grade for the final period are the attributes used in this study. The final grade was separated into two categories: a five-level grading system (A, B, C, D, E) and a binary grading system (pass and fail). With accuracy scores of 73.5% and 93.07% for the Portuguese dataset, the random forest classifier outperformed the five-level and binary grading systems. With the Mathematics lesson dataset, the J48 decision tree classifier outperformed with a five-level grading system accuracy score of 73.42%. In comparison, random forest had the highest binary grading system accuracy score of 91.39%. The WEKA tool was used for feature selection and data pre-processing. The wrapper method was used instead of the filter method because it provides better performance. The accuracy score of the classifiers improved dramatically because of this process. The random forest model has the highest overall accuracy score of 77.2% with the five-level grading system and 79.49% with the binary grading system on the Portuguese dataset. The J48 classifier had the highest accuracy of 79.49% with the five-level grading system on the mathematics dataset, while random forest outperformed with a binary grading system accuracy of 93.67%. When the wrapper approach is used, it can be deduced that the random forest classifier and binary grading scheme are the best overall among the others. Future work might use the filter subset selection approach or different machine learning classification algorithms on the model.

Sandoval et al. (2018) proposes an approach to at-risk student prediction that uses low-cost variables and an advanced algorithm. Low-cost variables in this context denote variables that do not necessitate active effort to collect data. There were 119,366 records left after reviewing the initial data from 21,314 undergraduate students. An LMS source, SAKAI, which stores each user's activity log, had 27,339,752 records, and an organisational information system, DARA, which stores students' demographic information and academic records, had 386,573 records. These data were collected over three semesters and then compiled into a database. The data was re-modified to get rid of all inconsistencies in the format. Due to the high dependence between instructor and courses, the CGPA and LMS usage variables were modified using a z-score transformation. This yielded a total of 36 variables, 25 from the LMS and 11 from the DARA. The final variables include gender, age, private/state funding, high school CGPA, language and mathematics score in university selection exams, university, career duration in semester, total course credits, number of semesters enrolled, university CGPA, resource, web and syllabus item accessed and downloaded, messages read, message composed and sent, message reply sent and forwarded, quiz submitted, assignment submission read, assignment revised, section read with lesson builder tool, forum topic accessed and response created, new chat message, poll vote entered and poll vote results, news feed accessed, wiki section read and page revised, profile preferences updated and added, profile privacy settings changed, personal preferences entry created, and final grade as the dependent variable. The data was further split into three classes. Large Courses was the first group, which included more than fifty students and at least one LMS interaction. There were 119,366 records found. The second group, Large Top 20%, is a subset of the first and contains data from the top 20% of large courses as well as average student LMS use. This resulted in a total of 19,232 records. The third and final group, Large Top 10%, is a subset of the second and includes data from the top 10% of large courses and average student LMS use. This had a total of 9,925 records. Linear regression (LR), robust linear regression (RLR), and random forest (RF) were the prediction algorithms used. A 10-fold cross-validation score, adjusted R2, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Average prediction Accuracy (APA), and Percentage Accurate Predictions (PAP) were the error metrics used. The most critical variables in the study were the students' school CGPA. The random forest classifier performed better than the rest, with an overall precision of 89.5% in the Large Top 10% and Large Top 20%, and 91.9% in the Large Courses category. The research is confined to the institution where it was conducted. It recommends that LMS systems be used in other educational institutions, as well as the use of other prediction algorithms such as SVM and neural networks.

Fernandes et al. (2019) used predictive analytics to analyse public school students' academic performance in Brazil to provide information to school administrators (teachers and guidance counsellors) to make effective educated decisions and appropriately allocate educational resources. The goal was to help third-year high school students increase their academic performance to graduate and pursue further education. The dataset was obtained from a State Department of Education repository of the Federal District of Brazil from 2015 to 2016. On 17 variables in the data collection, ranging from the students' personal, social, and geographic information to academic results, quantitative information on nominal data and the average numerical data were used. Two categories of datasets were used: the first (DS-I) contained variables collected before the academic year. The second (DS-II) includes academic variables collected two months into the semester. Descriptive analytics showed that the failure rate had risen from 12.5168% in 2015 to 13.0854% in 2016, and one of the main goals was to put in place measures that would prevent this from happening again. The data mining research followed a standard cross-industry standard method (CRISP-DM). A Gradient Boost Machine (GBM) classifier was used with the H2O algorithm and a boosting technique to improve the two classification models (CM-I and CM-II) on the two datasets. The 2015-2016 ROC curve for CM-I was 0.943076, and the validation data was 0.936658, indicating a strong rate of sensitivity success. The most critical variables that influence student academic performance are 'neighbourhood,' 'school,' 'city,' and 'age,' according to feature selection. In 2015-2016, the ROC value for CM-II was 0.989298, and after validation, it was 0.895314, which also indicated sensitivity success. Students' grades, subjects, and students' absence had a more substantial impact on variable selection than other variables. Other data mining techniques and the effect of other variables that can influence student results may be investigated in future studies. The research methodology can also be applied in the State Department of Education of the Federal District and higher education institutions.

To boost the accuracy of academic achievement prediction models, Zeineddine et al. (2021) suggests an automated machine learning approach. The data was gathered from several educational institutions in the United Arab Emirates. The program, school system, race, gender, age group, scholarship status, transfer status, acceptance on probation, student lives in the dorm, course load, Maths's level, and English level are among the predictor variables used. The students' result, a categorical variable depicted with a pass or fail, was the target variable. The data included 1014 students who were doing well and 477 students struggling, suggesting that the data was unbalanced. Synthetic Minority Oversampling Technique (SMOTE) was chosen to balance the dataset by adding extra data points to the training set to solve this problem. An ensemble model is a set of prediction algorithms that produces a higher level of accuracy. To build this model, the classifiers were chosen using the Auto-Weka search algorithm with the hyper-parameter optimisation option from the AutoML tool. Artificial Neural Network, K-Nearest Neighbours, K-Means Clustering, Naive Bayes, Support Vector Machine, Logistic Regression, and Decision Tree were chosen. The ensemble model is trained on 90% of the dataset and tested on the remaining 10% using a 10-fold cross-validation technique to determine accuracy. The kappa coefficient, which shows how well two variables agree, was also calculated. The ensemble model was the most accurate, with an overall accuracy of 75.9% and a kappa coefficient of 0.5, higher than other individual classifiers. This indicates that using an ensemble model with a balanced dataset would almost certainly result in more reliable results. Psychographic variables may be used in future studies to visualise the model's effectiveness. Students' career progress could also be predicted using auto-generated ensemble models.

Dien et al. (2020) takes a deep learning approach to student performance prediction. They also suggest data pre-processing strategies before feeding them to deep learning models. Can Thao University in Vietnam provided data from 16 academic units between 2007 and 2019. There are 3,828,879 records, 4,699 subjects, and 83,993 students in the data collected. After that, the data is pre-processed to delete any unwanted occurrence before being separated into training and testing sets. Data from 2007 to 2016 is used for training, while data from 2017 to 2019 is used for evaluation. Predictor variables are selected based on results from previous research, and they are: CGPA of courses passed in previous and preceded semesters, course ID, credits earned, English marks for three-level, entrance marks for three courses, entrance year, faculty, the field of study, GPA of courses in a preceded semester, high school, lecturer ID, mark recorded time, semester, number of credits, and student ID. Variables are transformed into an acceptable range using Quantile Transformation (QTF) and MinMaxScaler (MMS). The model was trained using two deep learning methods, convolutional neural network (CN1D) and Long Short Term Memory (LSTM), and a linear regression algorithm. The epoch patience for the deep learning models was set to 5, and the linear regression model was evaluated using RMSE and MAE. To boost efficiency even further, two optimiser functions, Adam and RMSprop, are applied to the models. Factor analysis with Pearson correlation coefficient reveals that CGPA and Course ID have the highest effect on the target variable, while Student ID had a negative impact. Other advanced methods may be used in future research to enhance performance.

Chen and Cui (2020) also, used LSTM to analyse student behaviour in a time-series format when using an LMS. In the LMS, dependencies between student regular click frequencies were used to generate time-series data. They went the extra mile by contrasting the LSTM network with eight other machine learning classifiers. The LMS data was collected over two semesters at a Canadian university. There were 141 and 527 students in the first and second semesters, respectively. The model was trained with 72% of the data from semester two and tested with the remaining 28%. The model is also designed to test all data samples from semester 1 to determine the model's generalisation. At-risk students were described as those who would receive a final grade of C+ or lower in this report. SMOTE was also used to deal with the training data's imbalance. Because of its indifference to unbalanced datasets, the Area Under the Curve (AUC) was used to evaluate LSTM and traditional classifiers. Neural Networks (NN), LR, Nave Bayes (NB), SVM, DT, KNN, RF, and gradient boosting machine (GBM) were the generic classifiers used in the analysis. Ten function variables relating to the use of the Moodle platform were identified, with the remaining 11 variables relating to modules used by course instructors. After that, weak variables were removed using Recursive Feature Elimination (RFE), resulting in seven final variables (five from RFE and two from other models). For hyper-parameter tuning, 10-fold cross-validation was used. Out of the other classifiers, NN, SVM, KNN, and RF performed the best overall, but the LSTM network still outperformed them all. The study was constrained by the fact that it only used a small dataset. Future analysis should look at how to deal with the class imbalance in the LSTM review.

## SUMMARY

Although extensive reviews have been carried out at different times using different indicators, there is still room for improvement and the development of more efficient models. The reviews of existing literature and methods have shown that the outcome of a prediction model hinges heavily on the data available and environmental factors in place in the institution. Hence, there would always be a need to explore the right combination of attributes that would accurately capture students' learning outcome.

# CHAPTER THREE

# RESEARCH DESIGN



## INTRODUCTION

This project aims to utilise data mining to realise various patterns in students' academic performance and develop an interactive web application that would counsel effective decision-making.

This chapter discusses the system's design and components and the methodology and computational techniques adopted for this project.

## REQUIREMENT ANALYSIS

The features and behaviour of a system and its operational constraints are outlined in its requirements. It identifies the required functionality to meet the needs of the customer. The requirements for software systems are divided into functional and non-functional requirements, which will be explored further down.



### Functional Requirement

Functional requirements define a system's fundamental behaviour, which essentially entails what it should and should not do. It specifies the services that the software system must provide. The system's functional requirements also stipulate how it should respond in various situations.

The functional requirements for the system entail:

1. The system shall provide a format for all data uploads.
2. The user shall be able to upload data of the student's results.
3. The system shall process and analyse uploaded data.
4. The user shall be able to interact with the system to generate an analysis of uploaded data.
5. The system shall provide suitable visualisations in a dashboard view.
6. The system shall provide information to support the decision-making process of the user.
7. The user shall be able to download the result of the data analysis.

### Non-Functional Requirement

Non-functional requirements are quality constraints the system must satisfy. They do not affect a system's core functionality because they will continue to perform essential functions even if they are not present. They are, however, required because they describe system behaviour, features, and general qualities that influence the user experience.

The non-functional requirements for the system entail:

1. The visualisation dashboards should be interactive.
2. The uploaded data must be in the correct format.
3. The website interface should be easy to navigate.

## SYSTEM ARCHITECTURE

System architecture entails the structure of a software system. It is a formal approach to expressing the system model in an easy-to-understand manner.

The architecture design provides a detailed understanding of how the user will interact with the system. It also visually depicts the system, demonstrating the various components that will make up the system. The system architecture consists of four fundamental elements:

1. **Database:** This is an organised collection of data that makes it simpler to access and analyse. It could include data from various sources, including company-wide data, data generated by various applications, and data gleaned from the internet. A decision support system's database could be a small database, a stand-alone system, or a huge data warehouse.
2. **Model:** The model consists of several mathematical and analytical models used to assess and produce the appropriate information from complex data—the model aids in executing analysis required for a specific sort of decision-making.
3. **Knowledge system:** This subsystem gives information on how data is related to one another. It organises information and presents decision-makers with different options for solving an issue. It also alerts decision-makers if there is a discrepancy between expected and actual results.
4. **Graphical User Interface (GUI):** The (GUI) makes it easier for the system and its users to communicate. It includes text, tables, charts, and illustrations to present the conclusions of the study. The user can select the best option for seeing the results based on his requirements.

Figure 3.1 shows the architecture of the system.

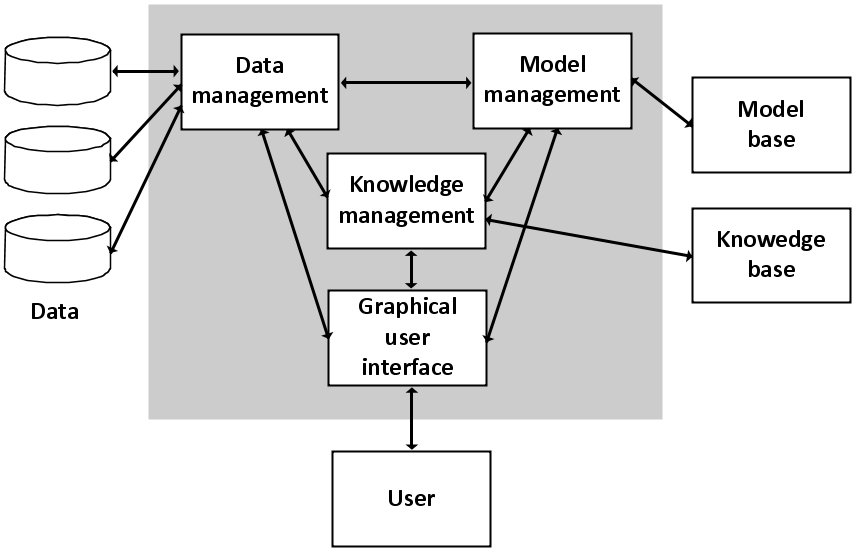


Figure 3.1 Architecture of the system (Hujer, 2011)

## SYSTEM DESIGN

The system design is based on the documented requirements. For this project, two specifications are required, which are the physical design and the logical design.



### Physical Design

The physical design represents the physical elements of your software system. It has to do with the actual input and output operations of the system. It focuses on how data is input into a system, verified, processed, and outputted.

It generates the functioning system by creating a design specification that specifies the functionality of the candidate system. User interfaces, procedures, and data are all part of its scope.

#### Input Design

The raw data that will be processed to produce output is known as input. The link between the information system and the user is the input design. A system input should be simple and easy to fill out, focusing on the user's attention. Input mechanisms facilitate the entry of data, both structured and unstructured, into the computer system.

The proposed system input design aims to capture accurate data from the users simply and straightforwardly. Textboxes and buttons are the primary input mechanisms employed by the system to capture information from the users

#### Output Design

A system's output design is critical because it guarantees that information is presented to users to comprehend it immediately. During output design, developers determine the required outputs and the necessary output controls and report layout prototypes. The contact with the device is strengthened by an efficient and intelligent output design, which improves user decision-making. The primary output necessary to satisfy the standards is determined, and techniques for presenting information and reporting system results are chosen when developing the program output.

### Logical Design

The logical design is a more abstract and intellectual approach to physical design. It does not concern itself with the physical implementation specifics but rather with the sorts of required data. The logical design process is organising data into a set of logical connections known as entities and attributes.

#### Use Case Diagram

A use case diagram is a behaviour diagram in Unified Modelling Language (UML) that helps to visualise the interactions between the system, other external systems, and various users of the system under development. Actors and use cases are used to model the functioning of a system in use case diagrams. The diagram depicts the system, as well as relevant use cases and actors, and connects them. A use case diagram does not specify the sequence in which use cases are executed. Still, it is helpful in visualising the system's functional requirements, influencing design decisions and development priorities. The use case diagram for the system is shown in Fig 3.2.

Diagram

Description automatically generated

Figure 3.2 Use Case diagram for the system

#### Activity Diagram

An activity diagram is a different variety of behaviour diagrams that depicts the control flow in a system and relates to the stages involved in executing a use case. It visually presents a series of actions in a system. It is also known as an advanced flowchart that depicts the flow of information from one activity to the next. The activity diagram is used to record a system's dynamic behaviour and build the executable system using forward and reverse engineering approaches. The activity diagram for the system is shown in Fig 3.3.

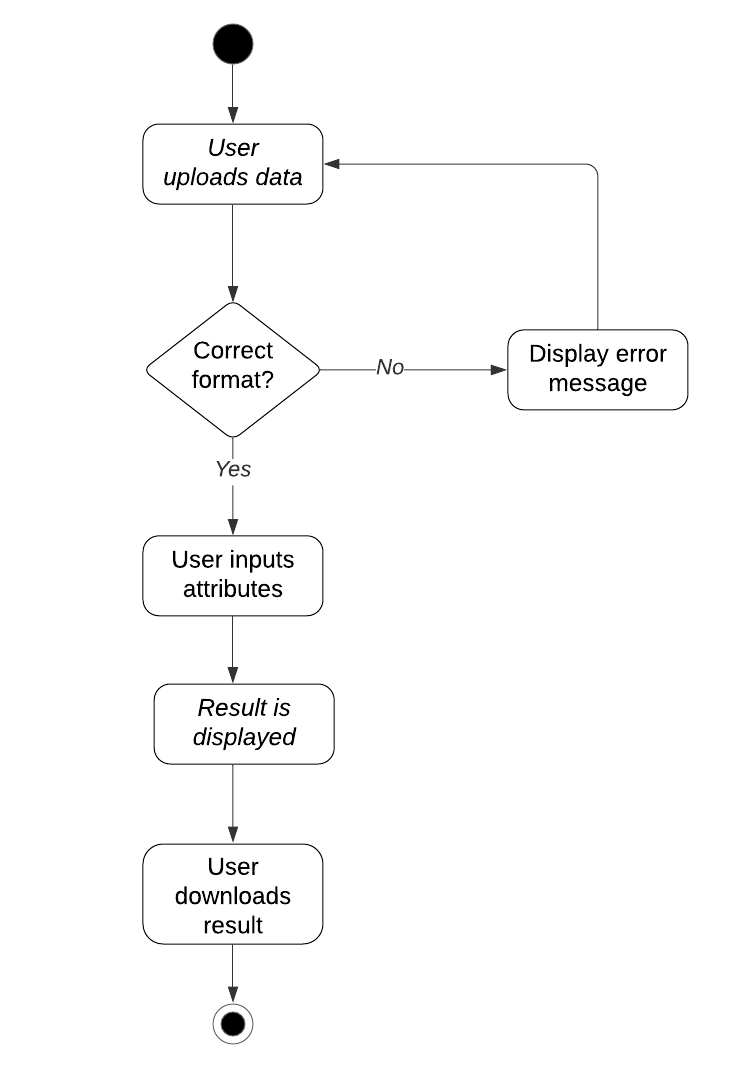


Figure 3.3 Activity diagram for the system

## RESEARCH METHODOLOGY

This section outlines in detail the research methodology workflow used in this project. First, it would outline the data collection process and then explain the data pre-processing and data analysis steps. Finally, the results would be interpreted. Figure 3.4 gives a graphical representation of the workflow.

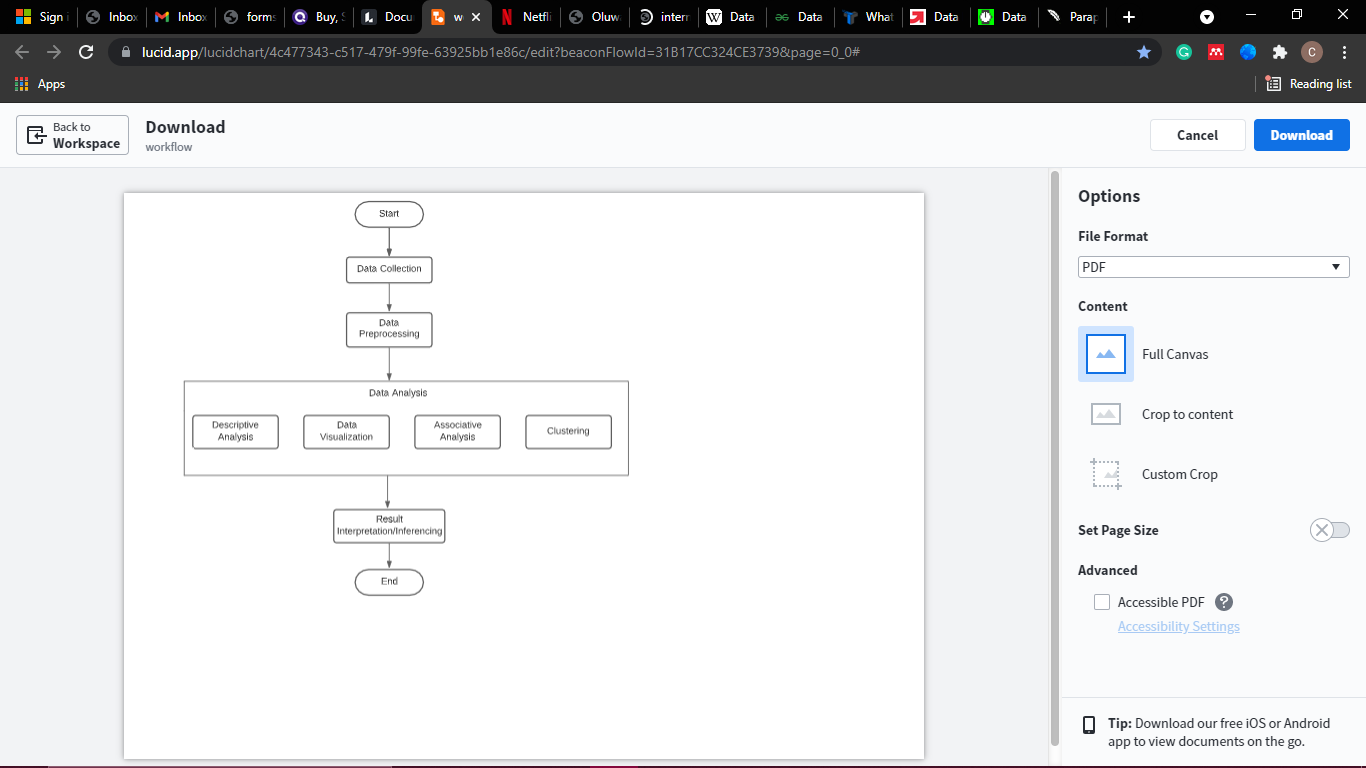


Figure 3.4 Research Methodology Workflow



### Data Collection

The data for this project was obtained from the Department of Computer and Information Science at Covenant University, Nigeria. The data included examination records from the first semester in the 2020/2021 academic session. The attributes included demographic information as well as students' biodata and their examination results. The features in the dataset are: matric number (which has been hashed to protect students' identity), number of failed courses (N FAIL), failed courses, incomplete courses, scores of students' in each course (represented by the course code, e.g. MAT 212), total course weight (TOT WTS), total unit (TOT UNIT), Grade Point Average (GPA), course weight from the previous semesters (PRE WTS), previous units (PRE UNIT), cumulative course weight (CUMWTS), cumulative course units (CUMUNIT), Cumulative Grade Point Average (CGPA), and class (1st, 2:1, 2:2, and 3rd class).

A questionnaire was also used in the analysis. The questionnaire was distributed to 50 final year students and lecturers of the department. The questionnaire aimed to get expert feedback on the factors that could affect a student's performance in a course. The factors considered for this analysis are course structure (hard, moderate, or easy), course nature (mathematical, theoretical, abstract, programming, or practical), and curriculum broadness (extensive, moderate, or short). Figure 3.5 shows a sample of the distributed questionnaire.

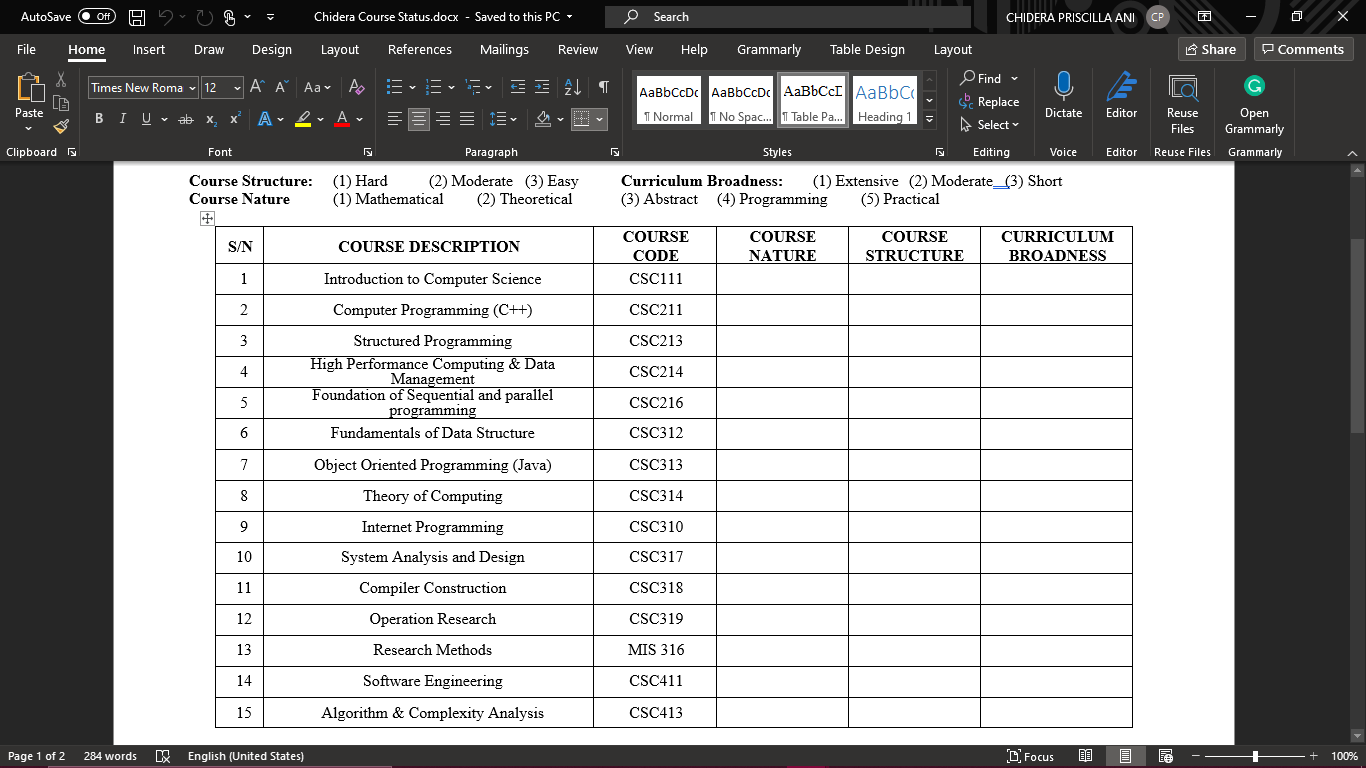


Figure 3.5 Sample questionnaire

Student capacity and the overall course performance was also taken into consideration. This data was obtained from the department based on the 2020/2021 academic session results. Student capacity deals with the number of students taking the course, and this was labelled low (less than or 15 students), average (between 16 and 70 students), or high (more than 70 students). The course performance attribute was based on the overall percentage of student's performance. For this analysis, course performance was 'good' if more than 50% of students got an 'A' or 'B' in the examination, and it was 'poor' if less than 15% got an 'A' or 'B'.

### Data Pre-Processing

Data Pre-processing involves transforming the raw dataset into a form that is well-formatted and ready to be used. It entails transforming the data into the format that is required for modelling. It converts the raw data into a readable and understandable form. Steps involved in this process include:

1. **Data Cleaning:** This involves correcting and removing inaccurate records from a dataset. Data cleaning often involves handling missing data, which occurs when no data is stored for a variable. Including missing values in the analysis might lead to inaccurate conclusions from the data. Handling missing values can be done by:
   * Ignoring the tuples, especially when the dataset is enormous, and the missing values are minimal.
   * Deleting a particular row where null values are more than 75%. This method is not very effective and is only recommended for a dataset with adequate samples.
   * Filling the missing values. This can be done by inputting the mean, median, or mode of the missing feature.

Missing data values in students' outcomes occurred in this study when no score was available because the student did not take the exam, which may have occurred for various reasons, such as the course being an elective. In this situation, missing data were replaced with zero to guarantee that the student's performance was unaffected.

1. **Data Editing:** this process involves reviewing and modifying acquired data, generally from a survey. It aids in the development of rules that eliminate data bias and provide consistent estimates, resulting in a credible analysis. This procedure can be carried out manually, with the use of a computer, or both.

Data was modified in Excel and Python for the purpose of this study. Hashing students' matriculation numbers to disguise identification, converting students' scores to grades, and defining baselines for student capability and course performance are some of the vital editing made throughout this project.

1. **Encoding Categorical Data:** Categorical data includes variables whose values are limited to a particular number of possible values, for example, gender, marital status. Some machine learning algorithms require only numerical values; hence it is necessary to encode these categorical variables to be included in the modelling phase and still retain their meaning.
2. **Data Transformation:** this involves changing the format and structure of the data. It is the process of converting data from its original form to the required format where it can be processed accordingly by the system or machine learning algorithm. To cluster students’ based on their performance, their scores were scaled to a defined range using Python’s sklearn StandardScaler() function.
3. **Feature scaling:** This step marks the end of data pre-processing. It is a method used to limit variables to the same range to ensure that no variable dominates another. It allows variables to be compared on common grounds. The most common techniques used for feature scaling are normalisation and standardisation.

### Data Analysis

Data analysis is a process of carefully examining the data to discover useful information. It involves applying statistical and logical techniques to describe and evaluate the data. For this study, the data analysis phase is divided into four steps which shall be discussed below.

#### Descriptive Analytics

Descriptive analytics or statistics describes raw data and turns it into an interpretable format. It describes past occurrences and behaviours and shows how they might influence future outcomes. Descriptive analytics does not go beyond the represented data; instead, it focuses on describing the available information at hand. To do this, descriptive analytics uses two techniques: data aggregation and data mining.

Data aggregation gathers all aspects of the data and represents it in a more summarised form which is easier for analysts to work it. At the same time, data mining searches the data for meaningful patterns and representations

This analysis was described using Pandas describe () method, which generates statistical information about a dataframe such as count, mean, standard deviation, minimum value, maximum value, and percentile.

#### Data Visualization

Data visualisation is a powerful tool for descriptive analytics. It helps to make sense of all the gathered data by providing a graphical representation of the data. Visualisation allows to quickly spot trends and patterns in the data that may not be visible when using charts or spreadsheets. Choosing the right visualisation tool is very beneficial for turning abstract data into something easier to understand. Visualisation tools used for this project include:

1. **Bar chart:** This is used to represent categorical data using rectangular bars where the length of the bars is proportional to the value they represent. The values usually describe a particular aggregation, such as sum. A bar chart could be vertical or horizontal.
2. **Box and whisker plot:** This is commonly referred to as a boxplot. It shows the distribution and skewness of values using the five-number summary: minimum, average, maximum, lower quartile, and upper quartile. They can be instrumental when comparing distributions between many groups and for visualising outliers.
3. **Scatter plots:** They are helpful for observe and show the relationship between numerical variables. It graphs pairs of numerical variables with one variable on each axis. If variables are correlated, points will fall along a curve or line.
4. **Pie Chart:** A pie chart uses pie slices to represent the relative magnitude of data in a circular graph. Each pie slice's arc length is proportionate to the amount it represents.
5. **Distribution Plot:** The distribution plot displays the distribution and range of numeric values plotted against a dimension. A normal distribution is usually referred to as a bell curve.
6. **Histogram:** A histogram organizes data points into specific ranges. It is similar to a bar graph, but it groups a range of outcomes into columns on the x-axis. The y-axis represents the count in the data for each column and can be used to visualize data distributions.

#### Association Mining

Association rule mining is a data mining technique used to find hidden relationships between attributes in a dataset (Banswal & Madaan, 2016). Association rules use If/Then statements to observe frequently occurring patterns or correlations from datasets. Association rules find the frequent itemsets then finds association rules from the generated itemsets. An association rule has two parts:

1. **Antecedent:** this is also known as the 'if' part of the statement. It is an item or group of items that are typically found in the itemset.
2. **Consequent:** this is also known as the 'else' part of the statement. It comes along as an item with an antecedent group.

The Apriori algorithm is a standard algorithm used for association rule mining. It was proposed by R. Agarwal and R. Srikant in 1994 and uses a bottom-up approach for frequent mining itemsets for Boolean association rules. The bottom-up approach implies k itemsets are used to generate k+1 itemsets. The algorithm begins by scanning the dataset to find the frequency of 1-itemsets based on user-defined minimum support. The frequency of 1-itemsets is used to locate the itemsets in 2-itemsets, which is then used to find 3-itemsets, and so on until no more k-itemsets are found. Any substantial subset of an itemset that is not frequent is likewise non-frequent (Al-Maolegi & Arkok, 2014).

Initial conditions for the algorithm include:

*Lk = set of large k-itemsets (set of items having minimum support); Ck = set of candidate k-itemsets (items to be counted); D = set of transactions, t ⊂ D.*

The algorithm is as follows:

*L1 = {frequent 1-itemsets};*

*for (k = 2;Lk−1〈〉0;k + + ){*

*Ck = set of new candidates;*

*for all transactions t ⊂ D*

*for all k-subsets m of t*

*if (m ⊂ Ck)m.count + +*

*Lk = {n ⊂ Ck∣n.count > = minsupp}*

*}*

*Set of all frequent itemsets = ∪ kLk;* (Aflori & Craus, 2007)



Figure 3.6 Diagrammatic representation of Apriori algorithm(Apriori Algorithm in Data Mining: Implementation With Examples)

Figure 3.6 shows a diagrammatic representation of the Apriori algorithm.

Standard terms associated with the Apriori algorithm include:

1. **Itemset:** a set of items that occur together.
2. **Frequent itemset:** this is an itemset that occurs frequently.
3. **Support:** the support defines the frequency of occurrence of an itemset in a dataset. Mathematically, support is the fraction of the total number of transactions in which the itemset occurs. An itemset with low support does not provide enough information on the relationship between items. For a rule X→Y, the support is given as:

|  |  |
| --- | --- |
|  | E*quation 3.1* |

1. **Confidence:** This metric determines how frequently a rule is valid. Given the antecedent, it is the conditional likelihood of the consequent occurring. It assesses a rule's dependability; however, it can be deceiving because the trust confidence in an association rule with a high frequency of consequence will always be high.

|  |  |
| --- | --- |
|  | *Equation 3.2* |

1. **Lift ratio:** Due to the tendency of the confidence metric to be biased to a frequent consequence, the lift ratio was introduced. Hence lift is the ratio of the observed support to that expected if the two rules were independent. A rule with a lift value close to 1 is completely independent, while a value greater than 1 indicates the rule has a high association.

|  |  |
| --- | --- |
|  | *Equation 3.3* |

#### Clustering

Clustering or cluster analysis is an unsupervised machine learning algorithm that involves grouping similar physical or abstract objects and dissimilar objects into different clusters. A cluster is a sub-group of similar objects, and each cluster has a centre called a centroid (Mahboob et al., 2020).

The k-means algorithm is the most widely used clustering algorithm due to its simplicity. The algorithm divides an unlabelled dataset into *k* clusters to have similar properties in the same cluster. The main aim of this algorithm is to minimise the sum of distances between the data point and their corresponding groups. It does this by minimising an objective function, in this case, a squared error function which is given by:

|  |  |
| --- | --- |
|  | *Equation 3.4* |

Where S is a K-cluster partition of the entity set represented by vectors yi (i ∈ I) in the M-dimensional feature space, consisting of non-empty non-overlapping clusters Sk, each with a centroid ck (k=1, 2, … K) (Kodinariya & Makwana, 2013).

K-Means is defined over continuous-valued data since it requires the ability to compute the mean. Calculating the distance or (dis)similarity between each pair of observations is required for classifying data points. A dissimilarity or distance matrix is the outcome of this algorithm. The Euclidean and Manhattan distances are two widely used methods for measuring cluster distance.

The k-means algorithm is given as follows:

1. Determine the centroid coordinate.
2. Determine the distance of each object to the centroids.
3. Assign each object to the group with the closest centroid.
4. After the assignment, recalculate the positions of the k centroids.
5. Repeat until convergence. (Mohamad & Usman, 2013)

The *k* in k-means defines the number of pre-defined clusters that need to be created in the process. Choosing a *k* value should be done carefully because different values of *k* produce different results. After the *k* value is selected, the data points nearest to the *k*-centre are assigned to a cluster. There are various ways to determine the number of clusters, *k* for an available dataset. Some commonly used methods include:

1. **Elbow Method:** This is by far the most used method. The elbow method looks at the total within-cluster sum of square (WSS) as a function of the number of clusters. The WSS score is a sum of Squared Errors of all the points which is given by:

|  |  |
| --- | --- |
|  | *Equation 3.5* |

The method begins with *k* = 2 and increments *k* at each step while calculating the clusters and the WSS value. It gets to a point where the WSS value drops dramatically for a particular *k* value. In a plot of WSS-versus-*k*, this is visible as an elbow. A problem with this method is that the 'elbow' cannot always be unambiguously identified. Figure 3.7 shows an example of the elbow method:

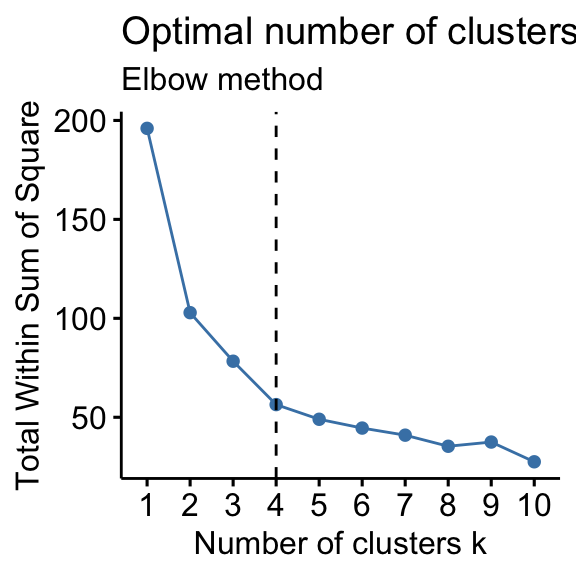


Figure 3.7 Example of elbow point

1. **Silhouette Method:** The silhouette coefficient method combines cohesion and resolution to measure how similar a data point is within a cluster instead of other groups. The silhouette coefficient is given as:

|  |  |
| --- | --- |
|  | *Equation 3.6* |

Where s(o) is the silhouette coefficient of the data point o, a(o) is the average distance between ‘o’ and all other data points in the cluster to which o belongs, and b(o) is the average distance from o to all clusters to which o does not belong.

The silhouette coefficient value is between [-1, 1]. A value of 1 indicates the data point is very compact in the cluster it belongs to and far from other clusters. A value of 0 indicates an overlap between clusters, and a negative value means data points have been assigned to the wrong clusters.

The silhouette coefficient method for determining *k* was implemented for the clustering analysis using sklearn’s silhouette score function.

1. **Gap Statistic Method:** The gap statistic compares the total within intra-cluster variation for different values of *k* with their expected values under the null reference distribution of the data. The estimate of the optimal clusters will be a value that maximises the gap statistic (i.e., that yields the most significant gap statistic). This means that the clustering structure is far away from the random uniform distribution of points. The basic idea of the gap statistics is to choose the number of *k*, where the most significant jump in within-cluster distance occurred, based on the overall behaviour of uniformly drawn samples.

Equation 3.7

Where Wk is total within intra-cluster variation and Wkb is the total within intra-cluster variation.

### Result Interpretation/ Inferencing

This last phase deals with interpreting all results obtained from the analysis and reaching a conclusion based on results. This phase is very vital as it sums up the research process and gives room for further works. Results from this analysis would be discussed in Chapter 4.

## CONCLUSION

The research design facilitates the smooth running of the different research activities and provides the most information with the least effort, time, and money. The system design process aims to give exact data and information about the system and its parts so that architectural entities described in system architecture models and views may be implemented. With all these pieces in place, you can create a fantastic system that integrates research processes.

# CHAPTER FOUR

# 4. SYSTEM IMPLEMENTATION



## INTRODUCTION

System Implementation makes a new system available to its users. This process defines how a system should be built and ensures that it meets the quality assurance needs of the users. This chapter would walk through the implementation of the proposed interactive system. It would also give an overview of the system requirements, data exploration and processing, libraries, programming languages and framework required to execute this project.

## SYSTEM REQUIREMENTS

System requirements encompass the necessary configurations that a system must have to execute the proposed system successfully.



### Minimum Hardware Requirements

Hardware requirements are the minimum physical components that are needed for a system. Table 4.1 shows the hardware requirements of the system.

Table 4.1 Hardware Requirements

|  |  |
| --- | --- |
| **Requirement** | **Hardware** |
| Processor | Intel® Core™ i5-7200U CPU@ 2.50GHz |
| Architecture | 64-bit OS |
| Primary Memory | 12.0 GB |
| Secondary Memory | 1 TB |

### Minimum Software Requirements

Software requirements include the minimum software that is needed to execute a system successfully. Table 4.1 shows the software requirements of the system.

Table 4.2 Software Requirements

|  |  |
| --- | --- |
| **Requirement** | **Software** |
| Operating System | Microsoft Windows |
| Programming Language | HTML, CSS, Python, JavaScript |
| Development Tool | Visual Studio Code, Jupyter Notebook |
| Web Server | Apache |
| Database Management System | MySQL |
| Internet Browser | Google Chrome |

## ANALYSIS RESULT

This section would discuss in detail the result of all analyses carried out in this research.



### Descriptive Analysis Result

Data exploration was carried out using Python libraries such as Pandas and NumPy. The first step of the analysis was to import the libraries and load the dataset. Figure 4.1 shows the dataset with 121 rows and 42 columns.

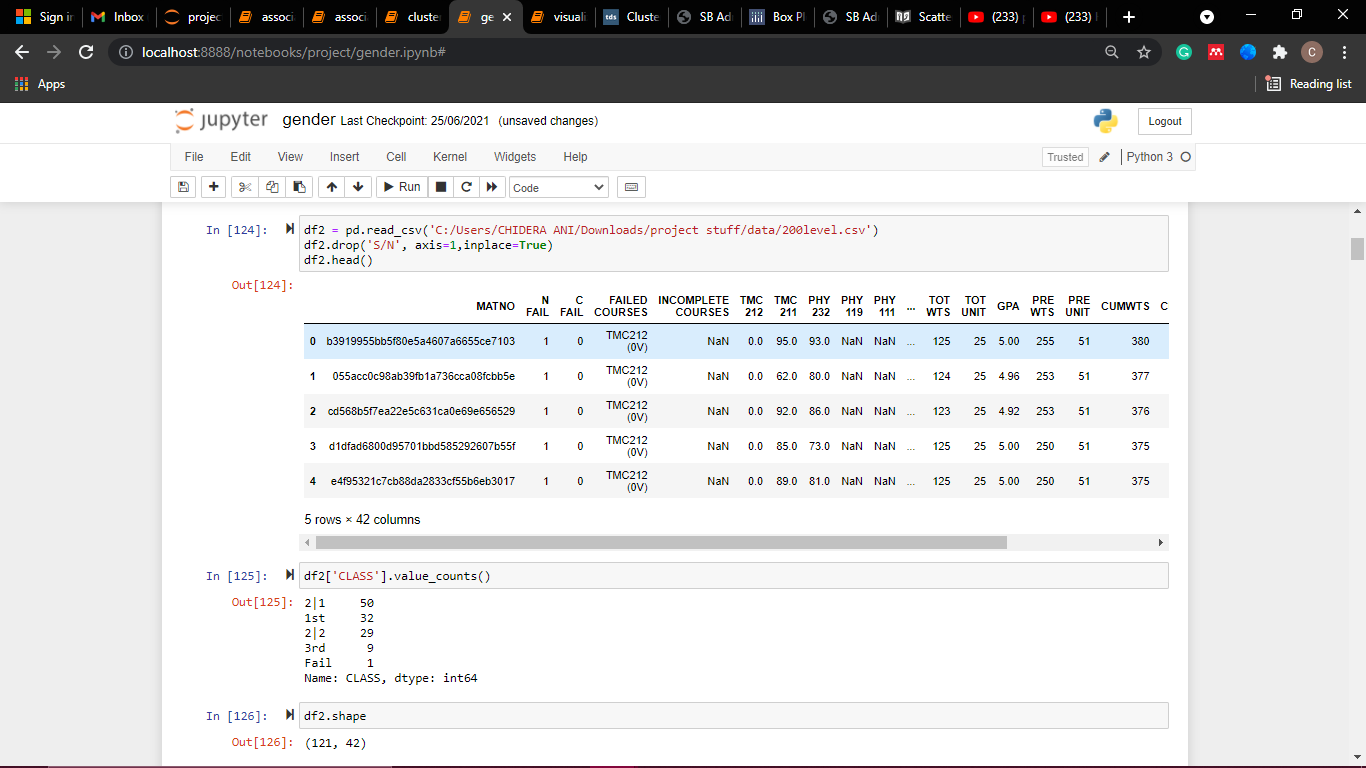
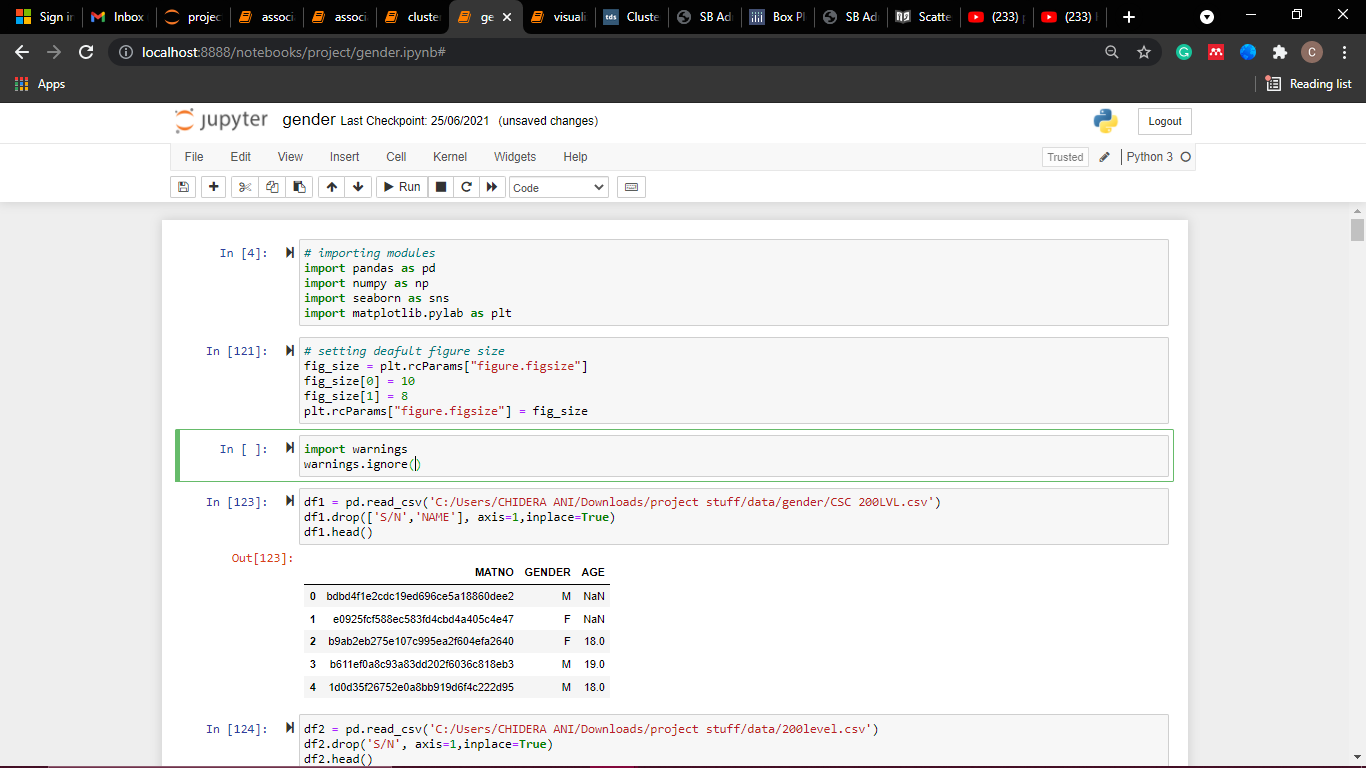


Figure 4.1 Importing libraries and loading dataset

Since not all students take the same courses, there were some missing values. These were replaced with 0 (zero) using the Pandas fillna() method. This can be seen in Figure 4.2.

A picture containing text, screenshot, computer

Description automatically generated

Figure 4.2 Replacing missing values

Pandas describe () method is used to view the descriptive analytics of the dataset. This method generates statistical details like count, mean, standard deviation, minimum value, maximum value, and percentile of a dataframe, as shown in Figure 4.3.

Graphical user interface, text

Description automatically generated

Figure 4.3 Descriptive analytics of dataset

To view the distribution of all attributes in the dataframe, the Pandas hist () function was used. It visualises the frequency and distribution of all variables in the dataset, as shown in Figure 4.4.

Chart, box and whisker chart

Description automatically generated

Chart

Description automatically generated with medium confidence

Figure 4.4 Distribution of variables

Figures 4.5 and 4.6 show the distribution of the CGPA variable.

Chart, histogram

Description automatically generated

Figure 4.5 Histplot of CGPA

Chart, histogram

Description automatically generated

Figure 4.6 Distribution of CGPA

To find the correlation between all variables and the CGPA, Pandas correlation matrix method, corr() was used, plotted with the Seaborn library as seen in Figure 4.7.

Chart

Description automatically generated with low confidence

Chart

Description automatically generated

Figure 4.7 Correlation between variables

### Data Visualisation Result

Data were visualized using Python’s Seaborn and Matplotlib visualisation libraries. Visualisation was used for two significant purposes, which would be discussed in this section.

#### Effect of Gender on Performance

To visualize the effect of gender on performance, the results of all levels in the department were joined together to give a more accurate analysis result. Visual plots like the bar chart and boxplot were used. Figure 4.8 shows the number of male and female students. It shows there are more male than female students.

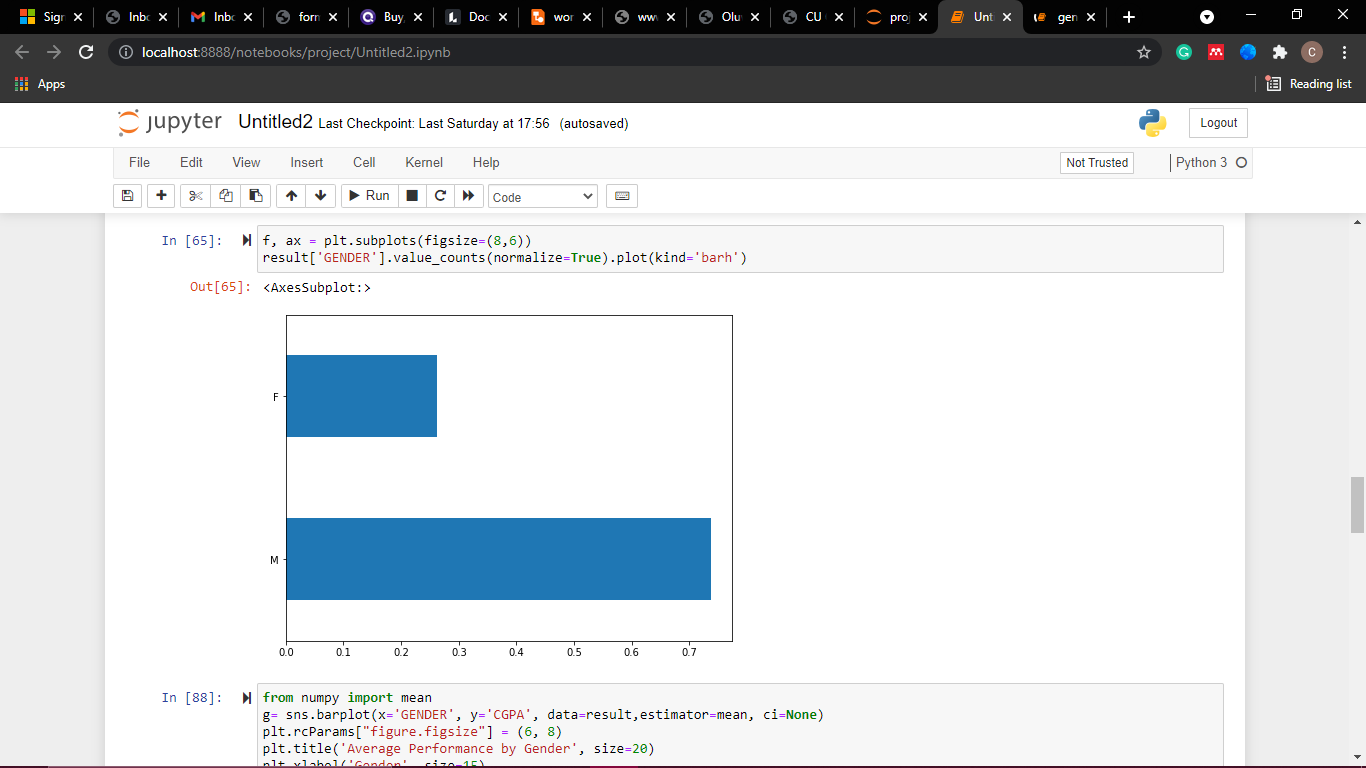


Figure 4.8 Frequency of male and female students

The CGPA distribution of male and female students is depicted in Figure 4.9. 75% of female students have a CGPA of 3.5 and above, which is considerably higher than male students. Figure 4.10 also buttresses this fact by showing that female students have a better average CGPA than male students.

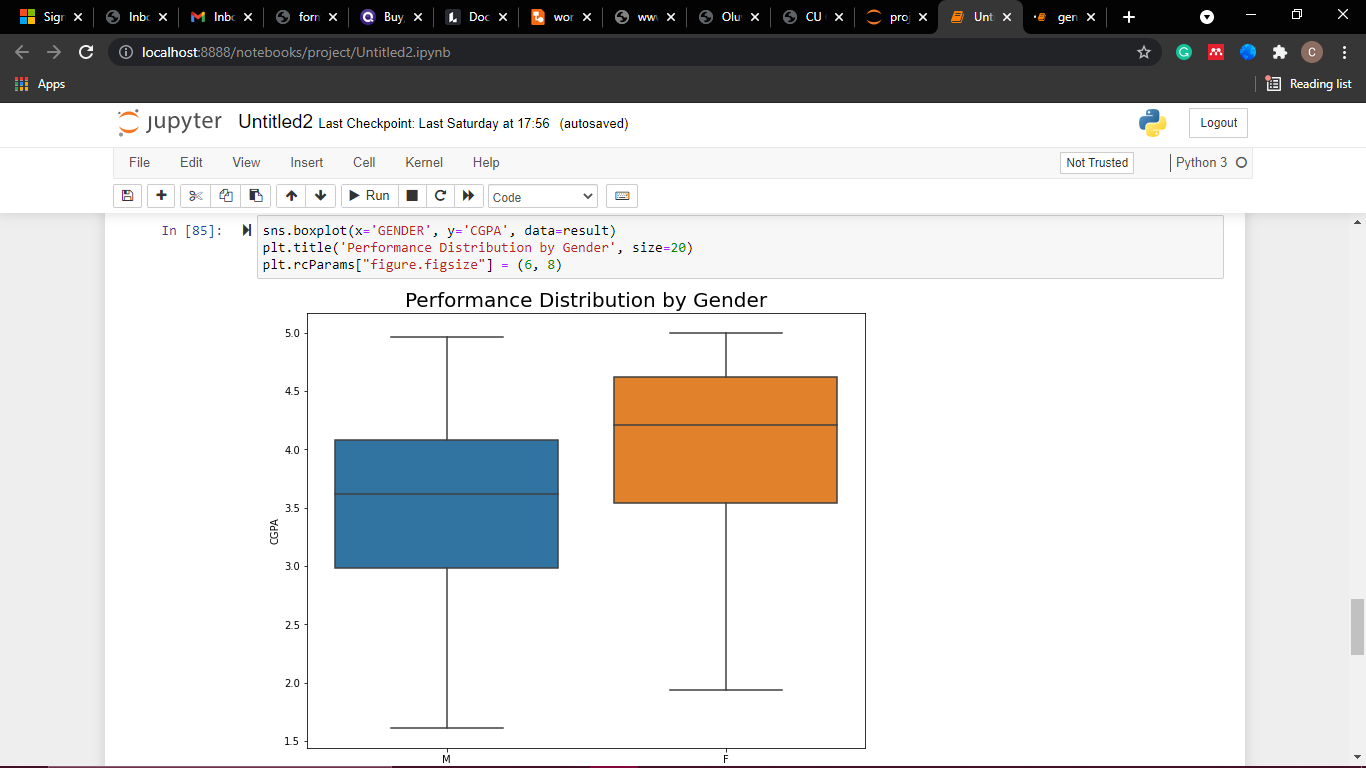


Figure 4.9 Distribution of male and female students by CGPA

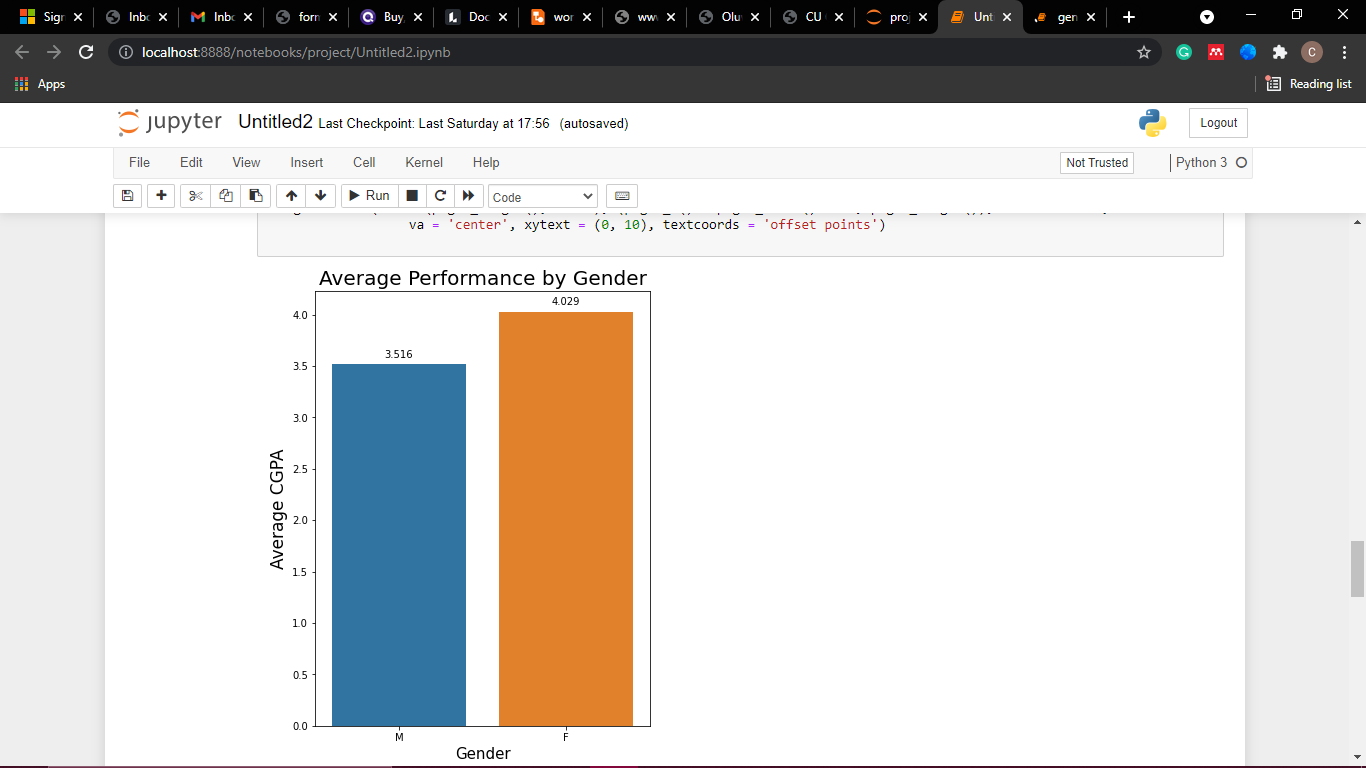


Figure 4.10 Average GPA performance by gender

#### Students’ General Performance

To investigate the general performance of students, visual plots like scatter plots, boxplots, and bar charts were used.

Figure 4.11 shows that majority of the students are in the second-class upper division.

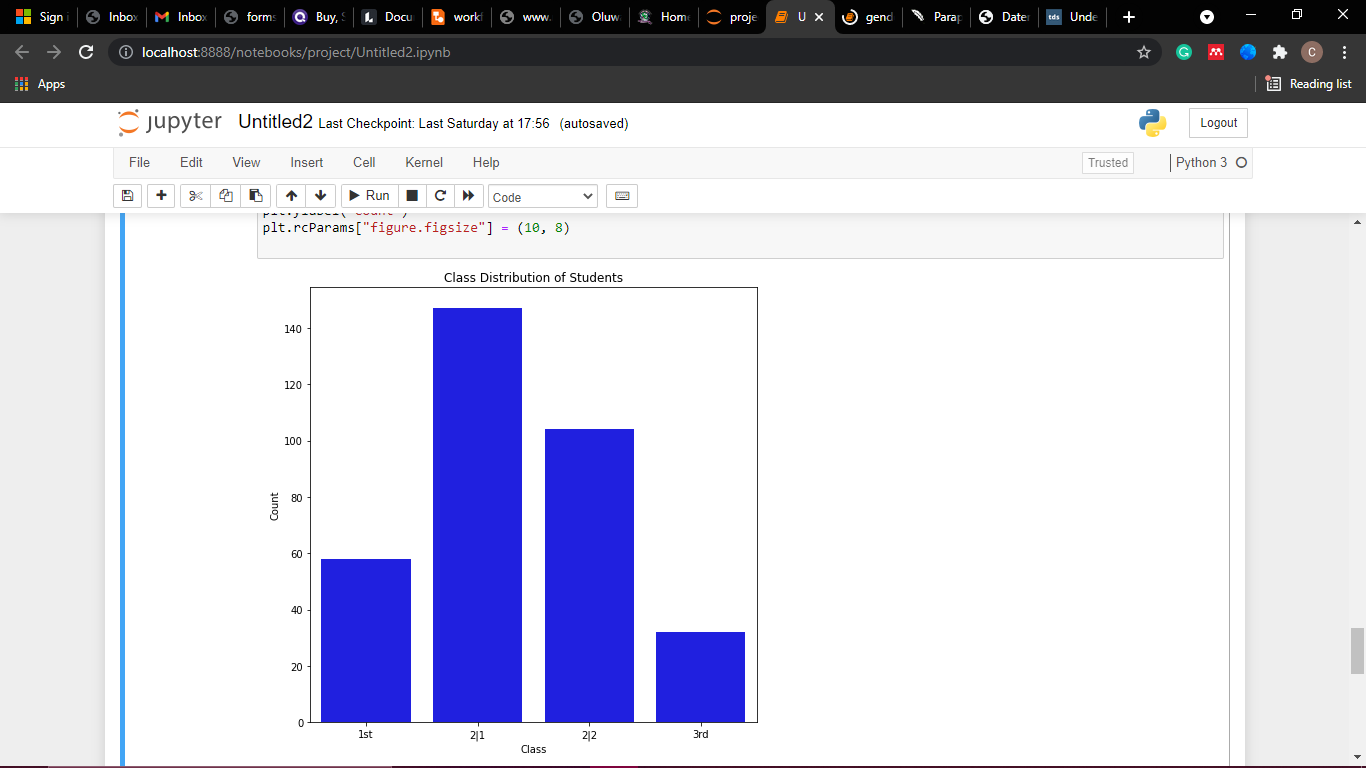


Figure 4.11 Class distribution of students

Figure 4.12 shows the relationship between students' previous and current CGPA and differentiates students that have increased and dropped in performance. Only very few students have improved in performance when compared to the previous semesters.

Chart, scatter chart

Description automatically generated

Figure 4.12 Current vs Previous CGPA

Figure 4.13 shows the distribution in performance of students in all courses. From this boxplot, courses, where performance is low, can be visualized, and administrative measures can be taken.

Chart, waterfall chart

Description automatically generated

Figure 4.13 Performance distribution of all courses

### Associative Analysis Result

Two sets of association mining were carried out. The first one aimed to find the relationship between students' performance in the various courses. The scores were converted to grades (i.e., A, B, C, D, F), and the course code was placed beside it (e.g., CSC111\_B). The data was loaded into Jupyter Notebook, as shown in Figure 4.14.

Graphical user interface, text, application

Description automatically generated

Figure 4.14 Loading in the dataset

The Apriori algorithm function by mlxtend (machine learning extensions), another Python machine learning library, only accepts binary data. Thus, the TransactionEncoder() function from mlxtend was used to convert the data to a binary format. This method encodes a Python list of lists into a one-hot encoded NumPy array. The data was then put back into a dataframe, as seen in Figure 4.15.

A picture containing text, screenshot, computer, indoor

Description automatically generated

Figure 4.15 One-hot encoding using TransactionEncoder()

The minimum support defined for this association rule was 0.05 (5%) with a confidence of 0.1 (10%). These low values were chosen so that users would input the desired course and grades and get the associated support and confidence when the model is loaded into the website application. The max\_len attribute, which allows limiting the number of rules was set to 2 since users would only be able to input a maximum of two courses. This resulted in 811 rules. The frequent itemsets can be seen in Figure 4.16, and the generated support, confidence, and lift values are shown in Figure 4.17.

A picture containing text, screenshot, computer

Description automatically generated

Figure 4.16 Generating the itemset

Graphical user interface

Description automatically generated with medium confidence

Figure 4.17 The generated rules

The next association rule mining was carried out using data from the questionnaire. This rule mining aims to determine the effect of a course's nature, structure, curriculum broadness, and student capacity on students' performance. To do this, data obtained from the questionnaire was input into Excel in a binary format. Figure 4.18 shows the dataset after it has been loaded into Jupyter Notebook.

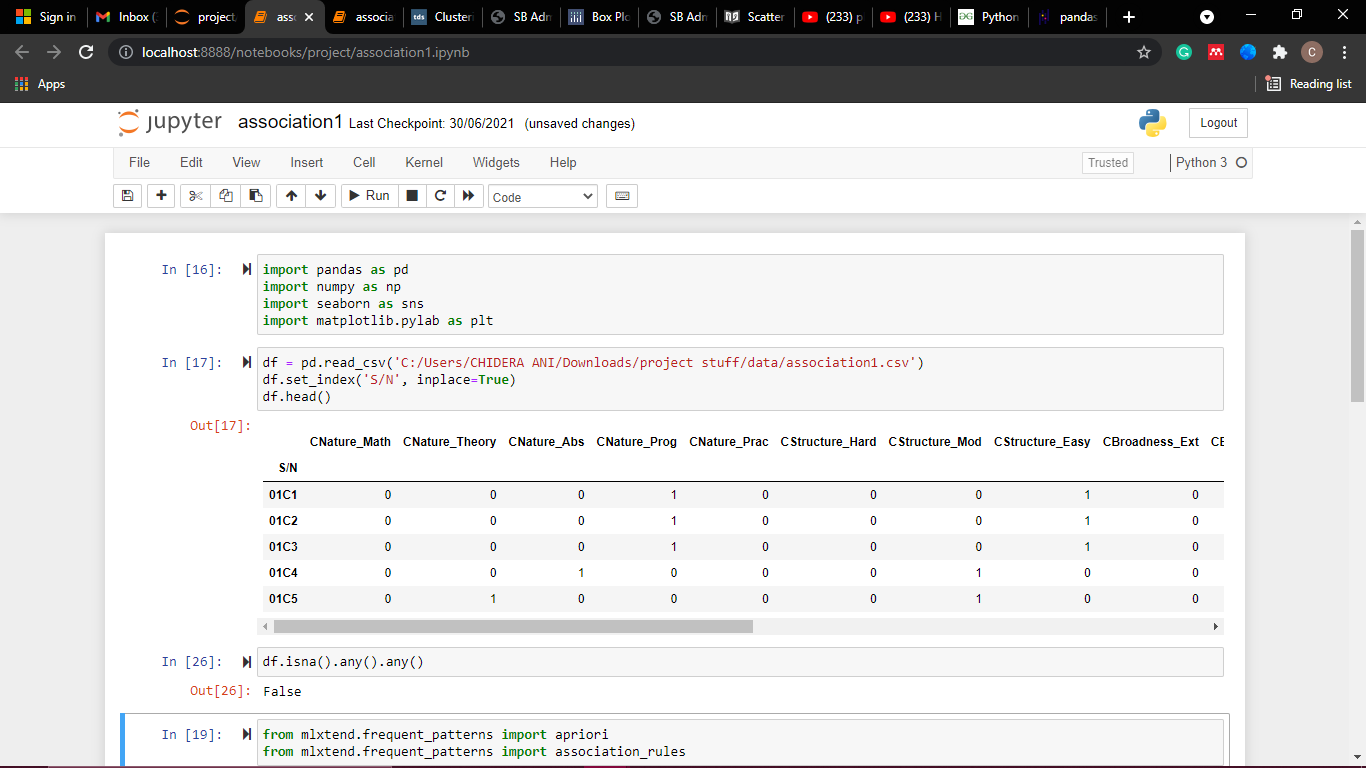


Figure 4.18 Loading the dataset

Minimum support of 0.2 and confidence of 0.6 was defined, and the generated rules are shown in Figure 4.19.

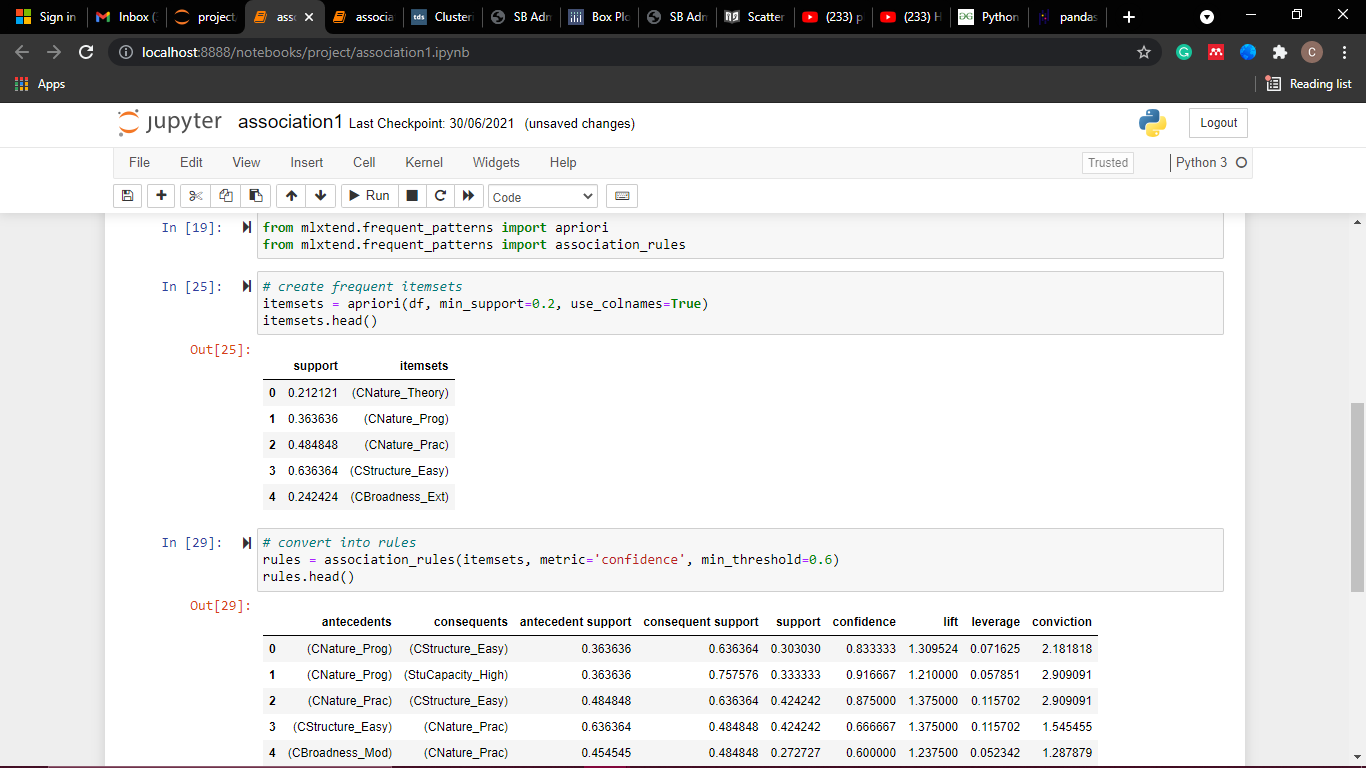


Figure 4.19 Generating itemset and rules

The rules were filtered to include only the course performance variables in the consequent column (CPerformance\_Good, CPerformance\_Fair, and CPerformance\_Poor), as seen in Figure 4.20.

A picture containing text, screenshot, computer, indoor

Description automatically generated

Figure 4.20 Resulting rules

### Clustering Result

The clustering analysis carried out in the project aimed to group students based on their raw scores in various courses. The traditional approach of grouping students in Nigerian universities is by class (e.g., the second-class upper division) which could be somewhat biased. The method implemented here groups students using their actual scores and implies a more accurate approach. The columns that contain students' scores are placed in a new dataframe, as shown in Figure 4.21.

Calendar

Description automatically generated with medium confidence

Figure 4.21 Clustering dataframe

The columns are scaled using sklearn's StandardScaler() method, which standardises features by removing the mean and scaling to unit variance. It transforms the data such that its distribution will have a mean value of 0 and a standard deviation of 1. This is shown in Figure 4.22.

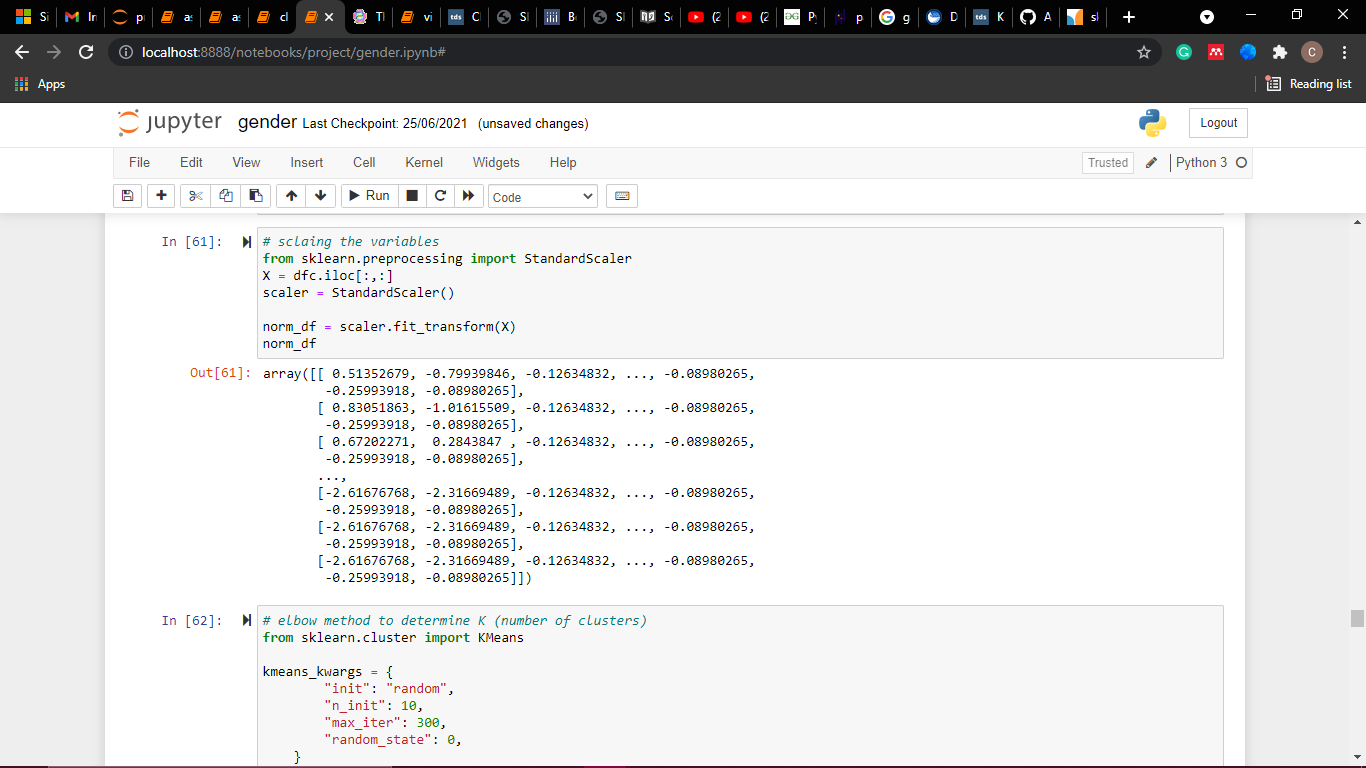


Figure 4.22 Scaling variables

The elbow method, silhouette coefficient, and gap statistics methods were used to determine the number of clusters. These are all implemented using the sklearn library, as shown in figures 4.24, 4.26, and 4.28. The elbow method and gap statistics method suggest 3 clusters, while the silhouette method suggests 2 clusters.

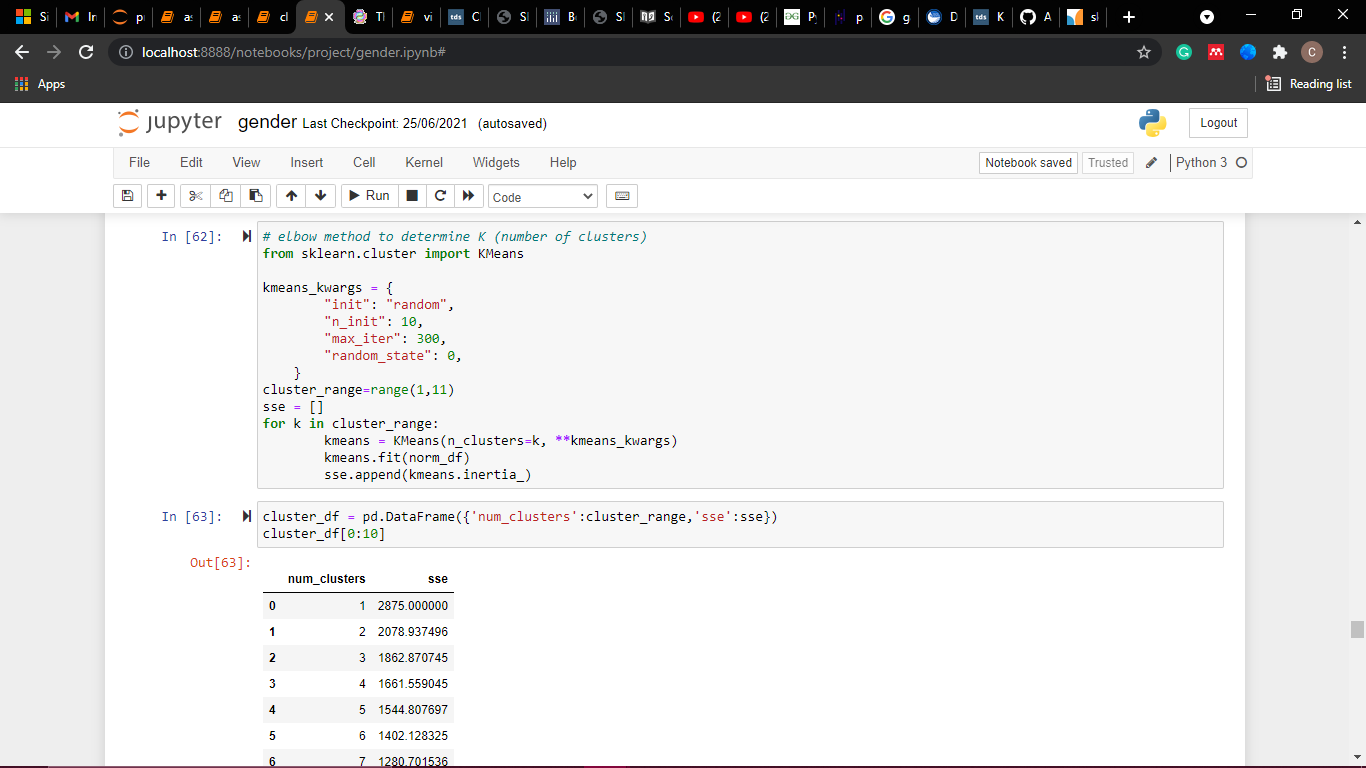


Figure 4.23 Implementing the elbow method

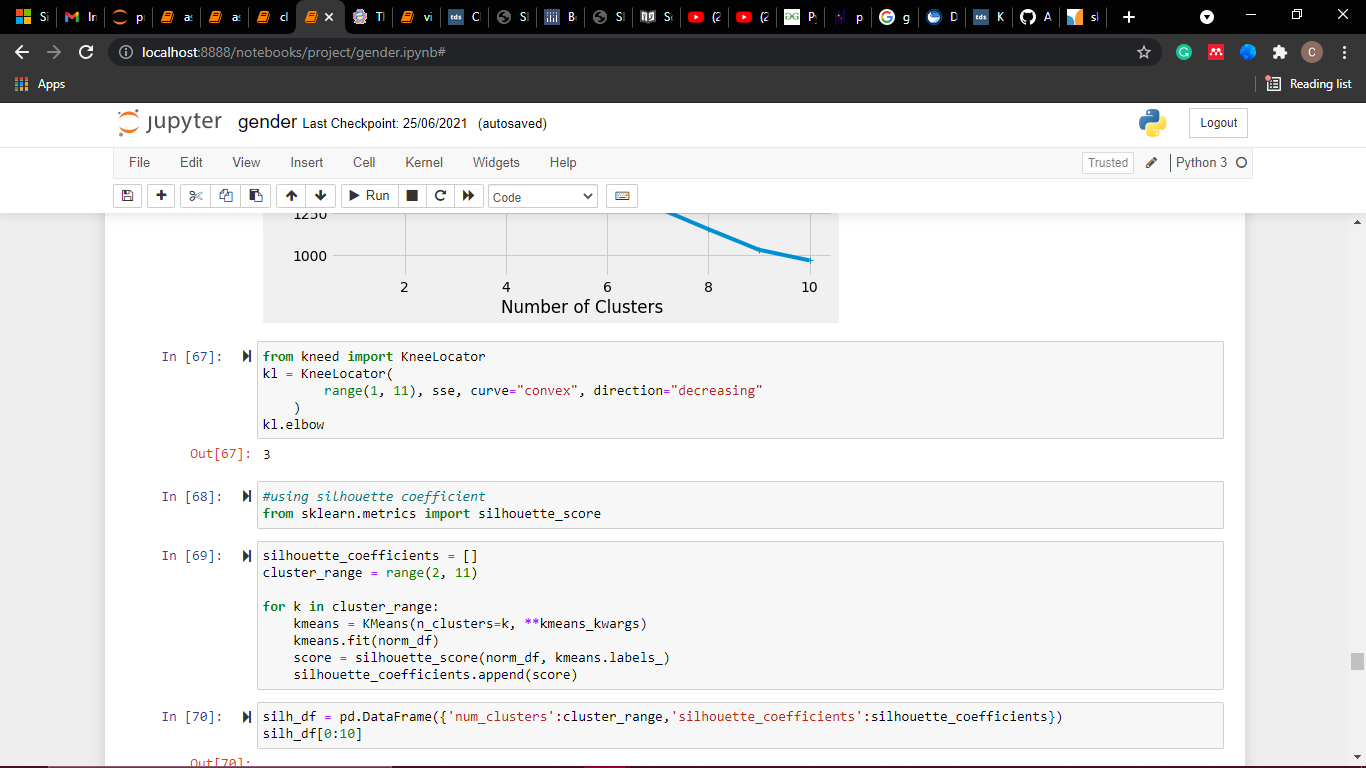
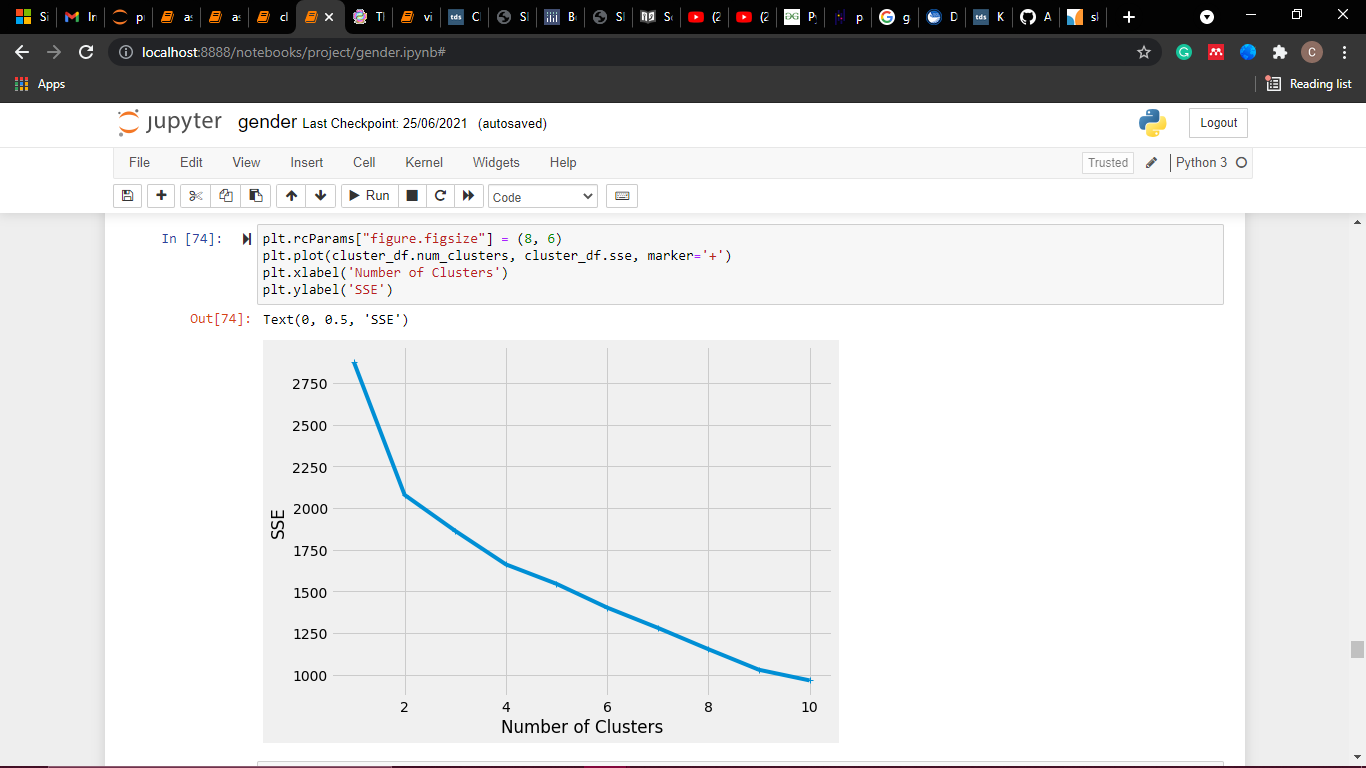


Figure 4.24 Result of using the elbow method

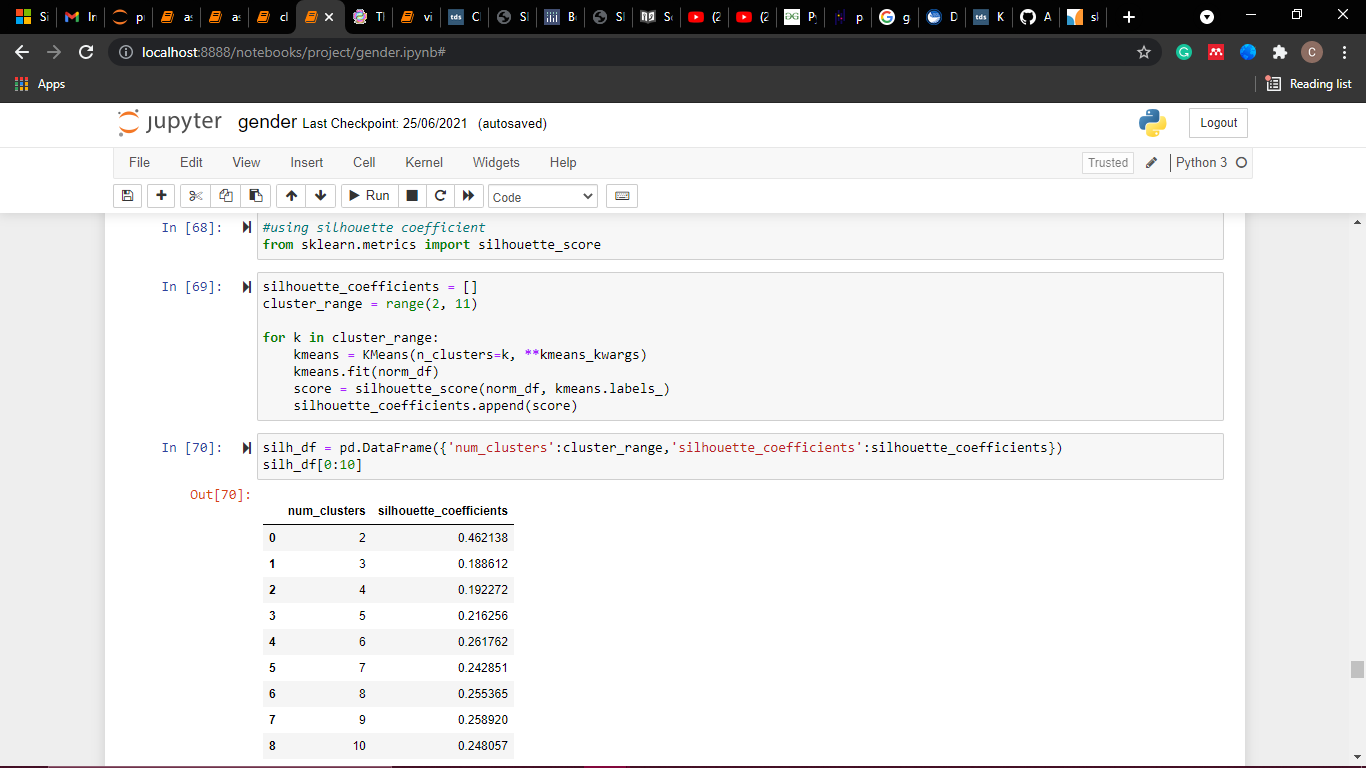


Figure 4.25 Implementing silhouette coefficient

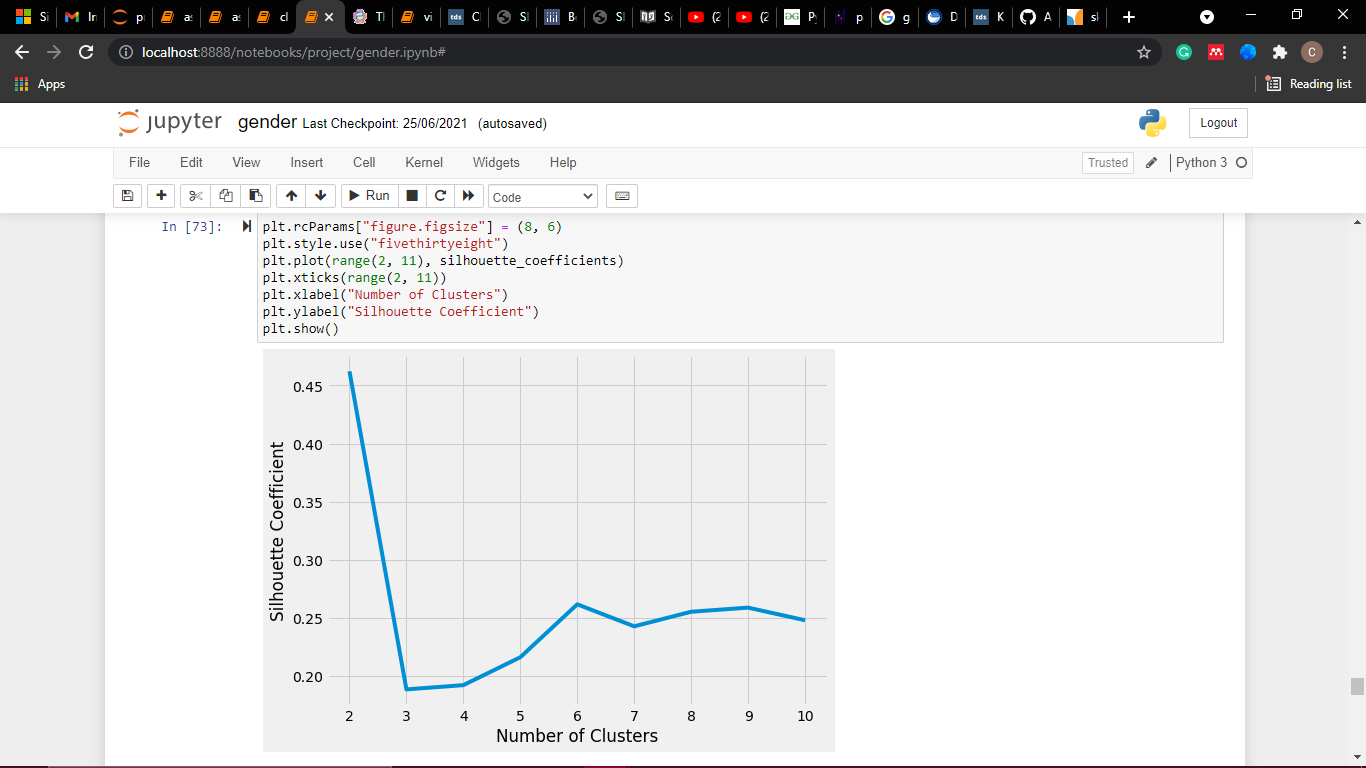


Figure 4.26 Result of using silhouette coefficient

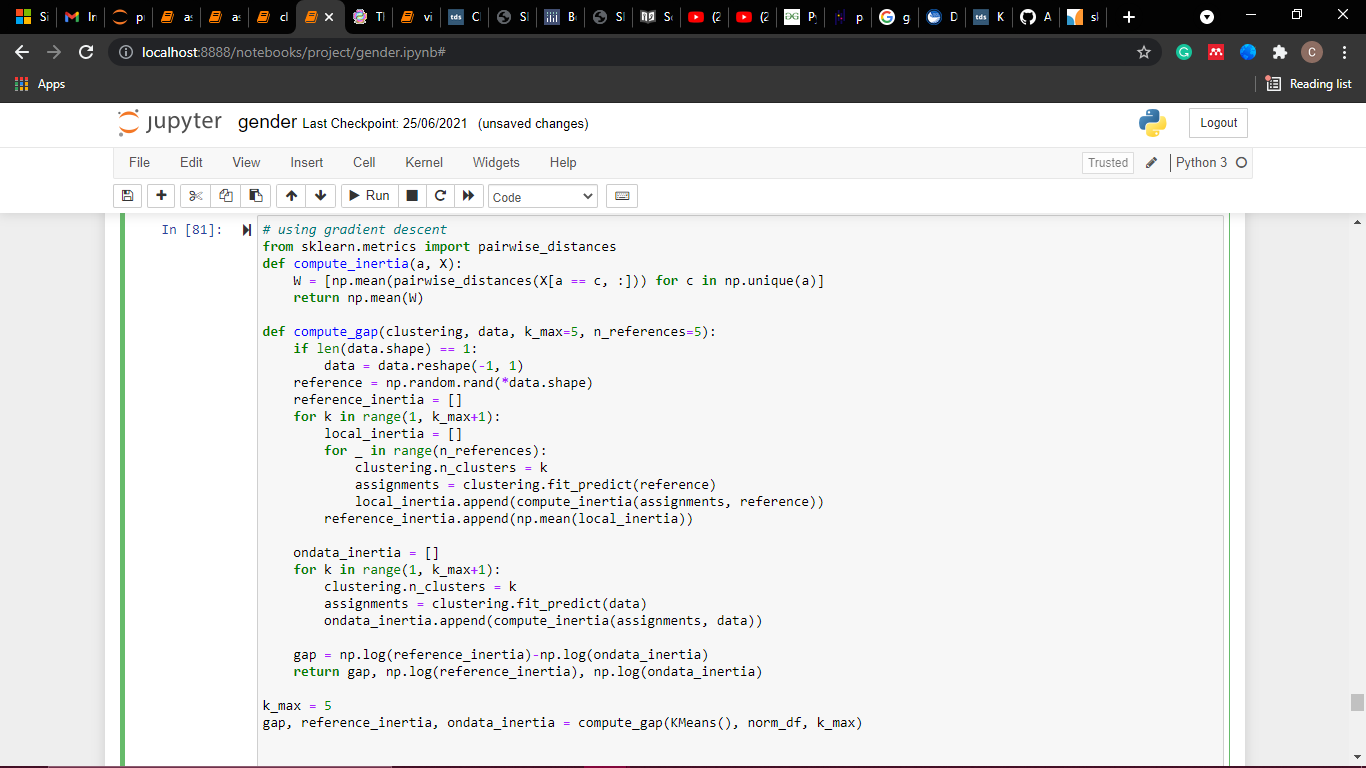


Figure 4.27 Implementing gap statistics

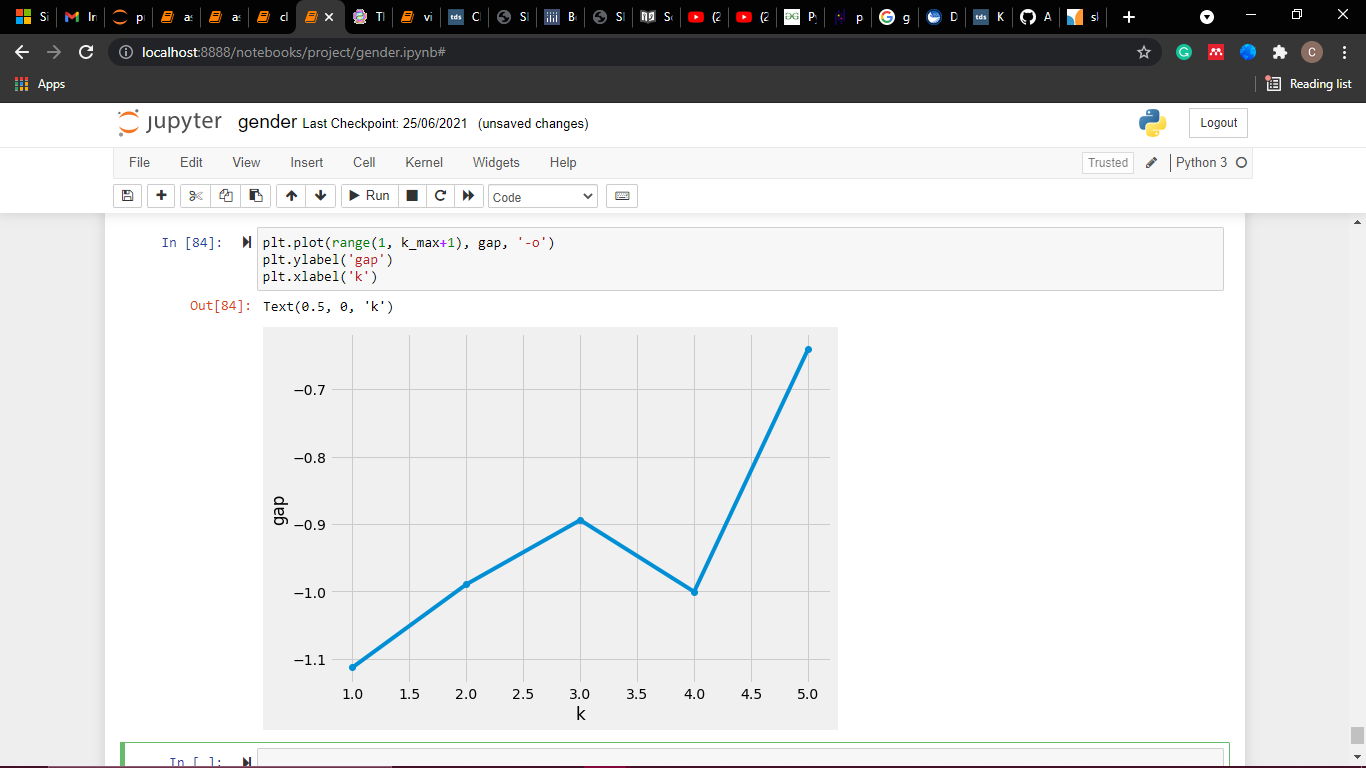


Figure 4.28 Result of using gap statistics

Next, the sklearn library is used to perform the clustering. This can be visualised in figure 4.29.

Chart, scatter chart

Description automatically generated

Figure 4.29 Clustering the variables

The number of students in each cluster is shown in Figure 4.30.

A picture containing text, screenshot, computer

Description automatically generated

Figure 4.30 Number of students in each cluster

Cluster labels are added to the dataframe, as seen in Figure 4.31.

A picture containing text, screenshot, computer, indoor

Description automatically generated

Figure 4.31 Adding cluster labels to the dataframe

From the profiling chart shown in Figure 4.32, cluster 2 (in blue) has the highest performing students, cluster 0 (in green) has the average students, and cluster 1 (in red) has the lowest-performing students.

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

Figure 4.32 Profiling chart from clustering

## RESULT INTERPRETATION

From the analysis carried out, the following results were obtained. First, visual depictions show that females perform better on average than males. Second, the average performance of students this semester dropped in comparison to the previous semesters. Hence, administrative measures should be carried out to determine the reason behind that. Finally, association mining results revealed that a moderate curriculum, a practical course, fair student capacity, and a moderate course structure are the factors that lead to a good course overall performance.

## PROGRAM MODULES AND INTERFACES

This section gives details about each page that is available in the website application.



### Home Page

The home page is the landing page that users view immediately they open the application. It gives a brief overview of the system and other web pages available. It also helps the user navigate easily through the system. Figure 4.33 shows the home page.

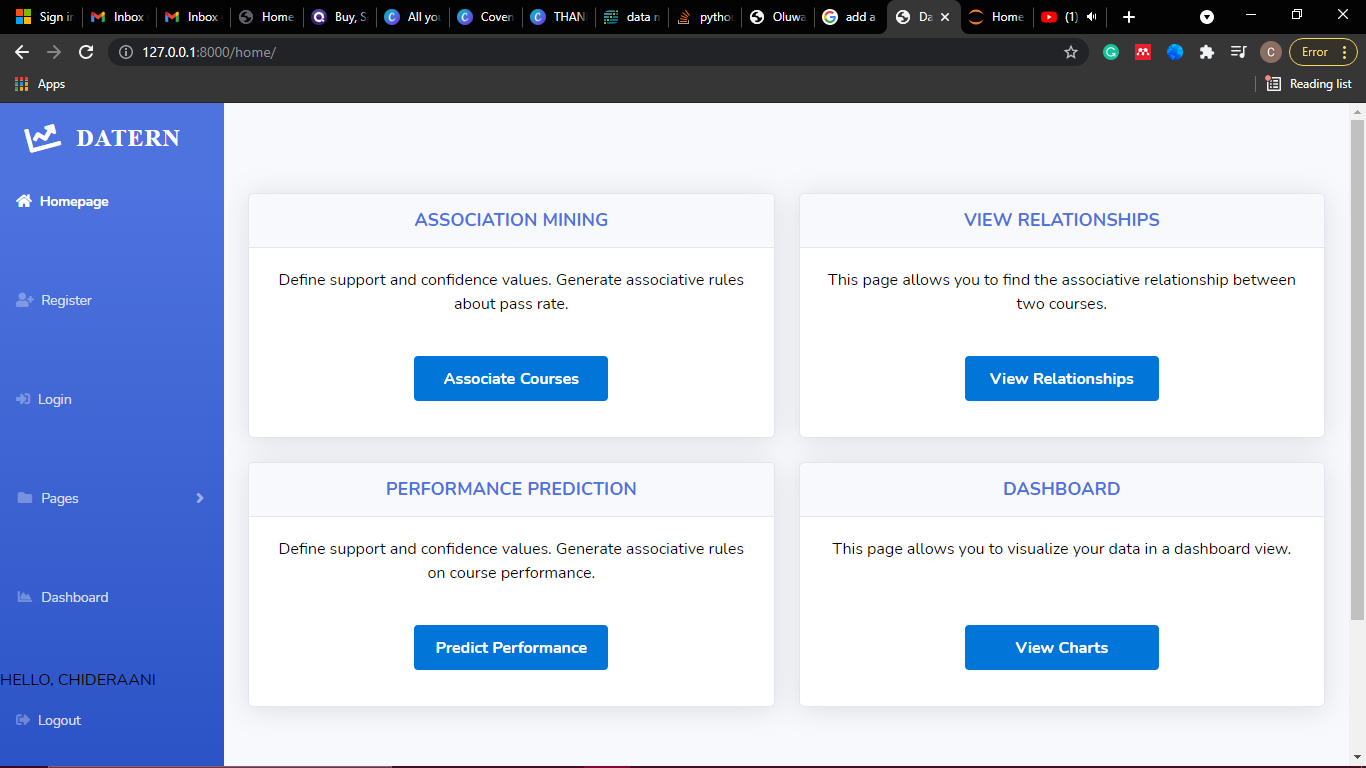


Figure 4.33 Home Page



### Registration Page

On this page, new users must create an account to be able to interact with the system. They would be required to input their first name, last name, email, and password, which would be saved into the database. Figure 4.34 shows the registration page.

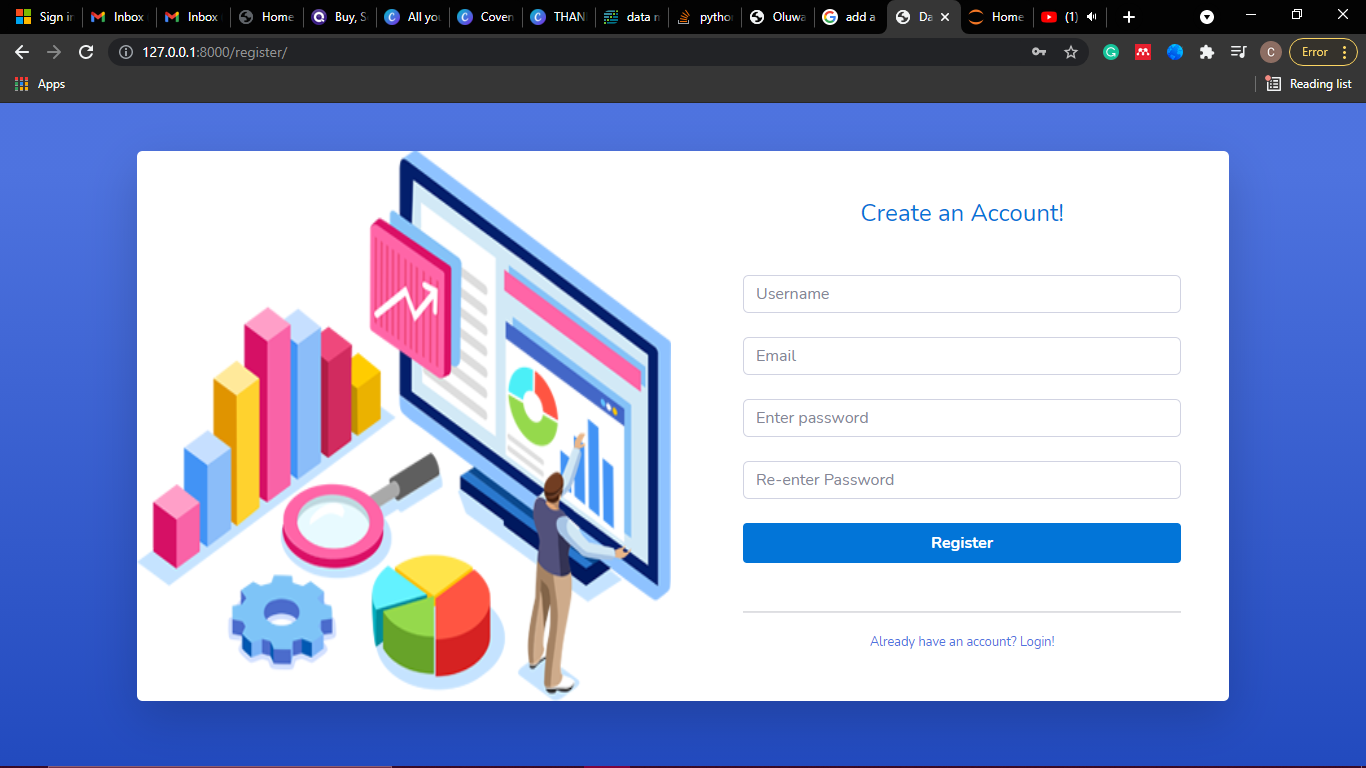


Figure 4.34 Registration Page

### Login Page

The login page allows existing users into the system. They would be required to input their username and password, which would be validated via the database. Figure 4.35 shows the login page.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 4.35 Login Page

### View Association Page

This page specifies a format for file upload, allows users to input a file, and returns an error message if the file format is wrong. Users specify desired minimum support and confidence and can download the generated rules. Figure 4.36 shows the association page.

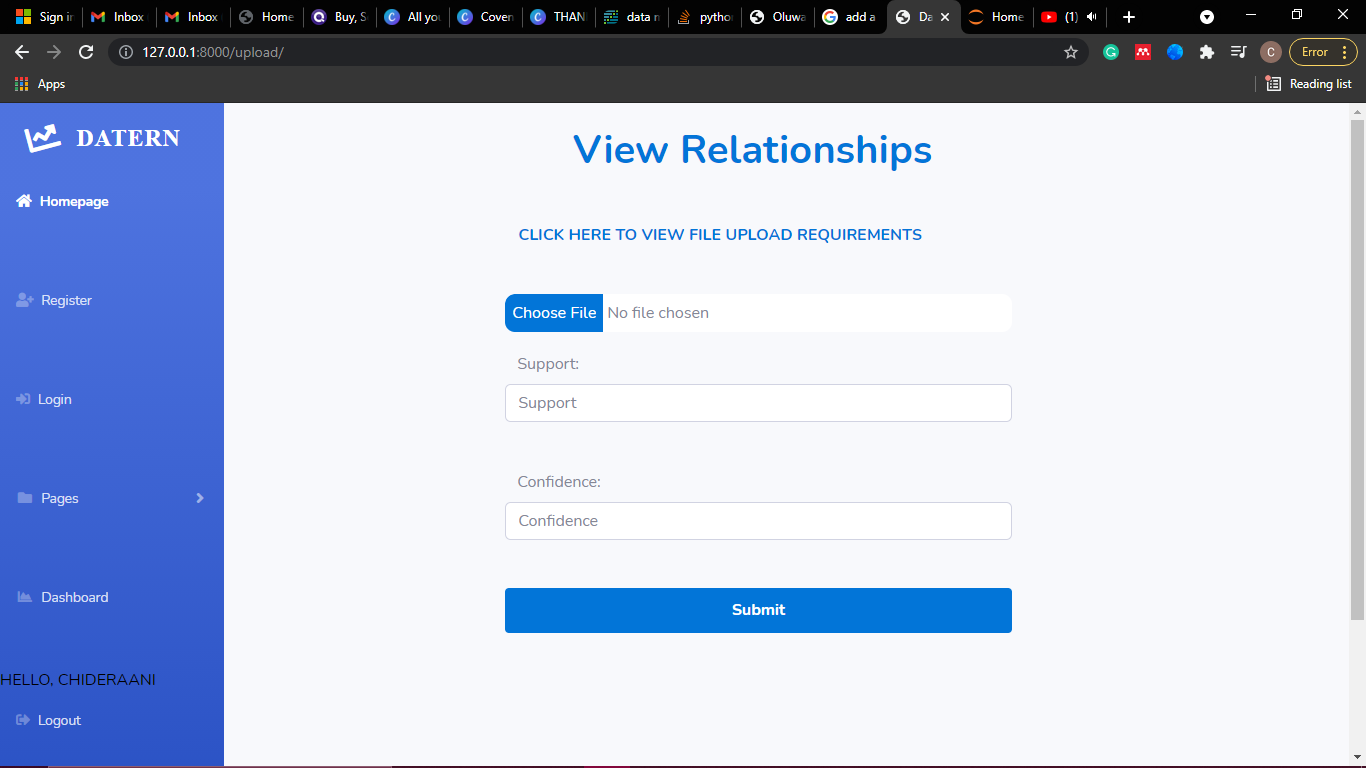


Figure 4.36 View Association Page

### Course Association Mining

On this page, users also upload files based on a defined format. They input desired courses and grades and can view the relationship between the courses. Figure 4.37 shows the course association mining page.

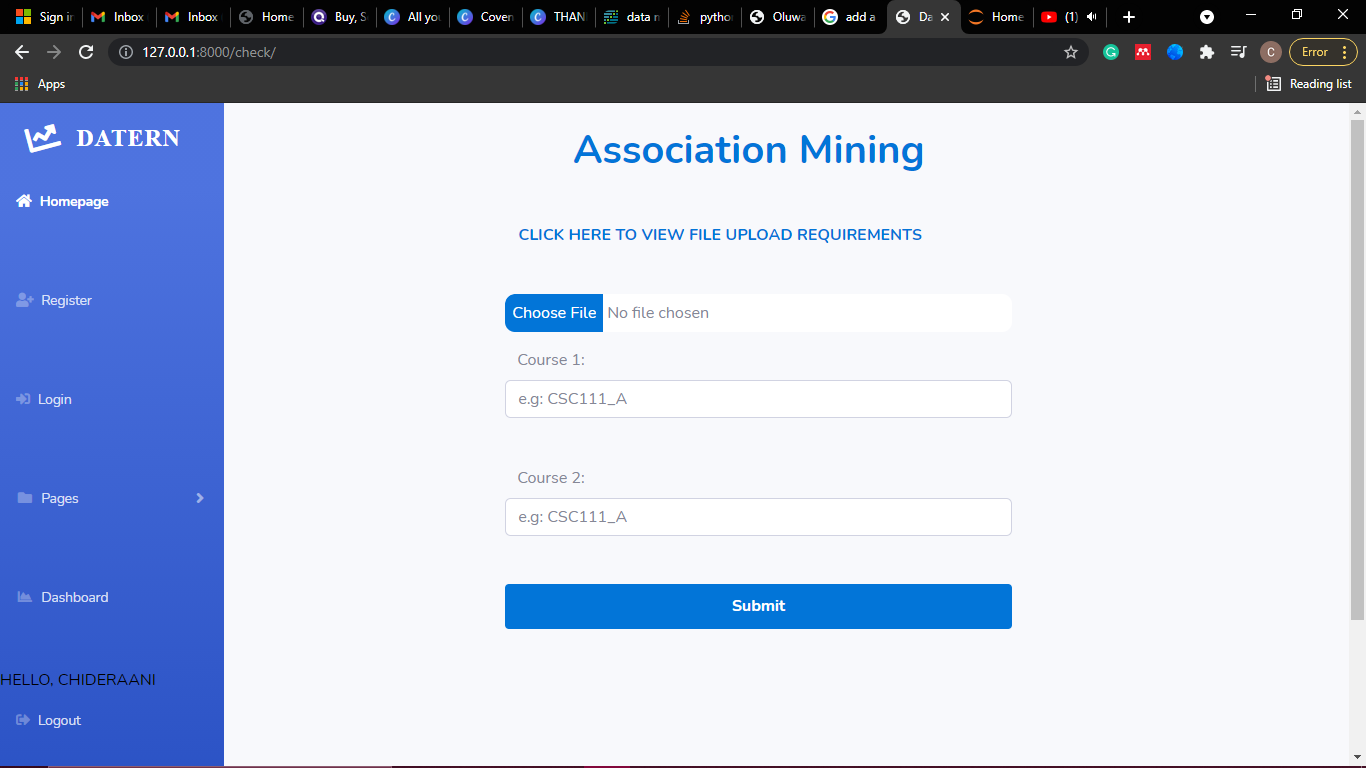


Figure 4.37 Course Association Mining

### Course Performance Prediction Page

This page generates association rules based on the questionnaire data. Users can upload their file, specify support and confidence values, and download the resulting association rules on course performance factors. Figure 4.38 shows the course performance prediction page.

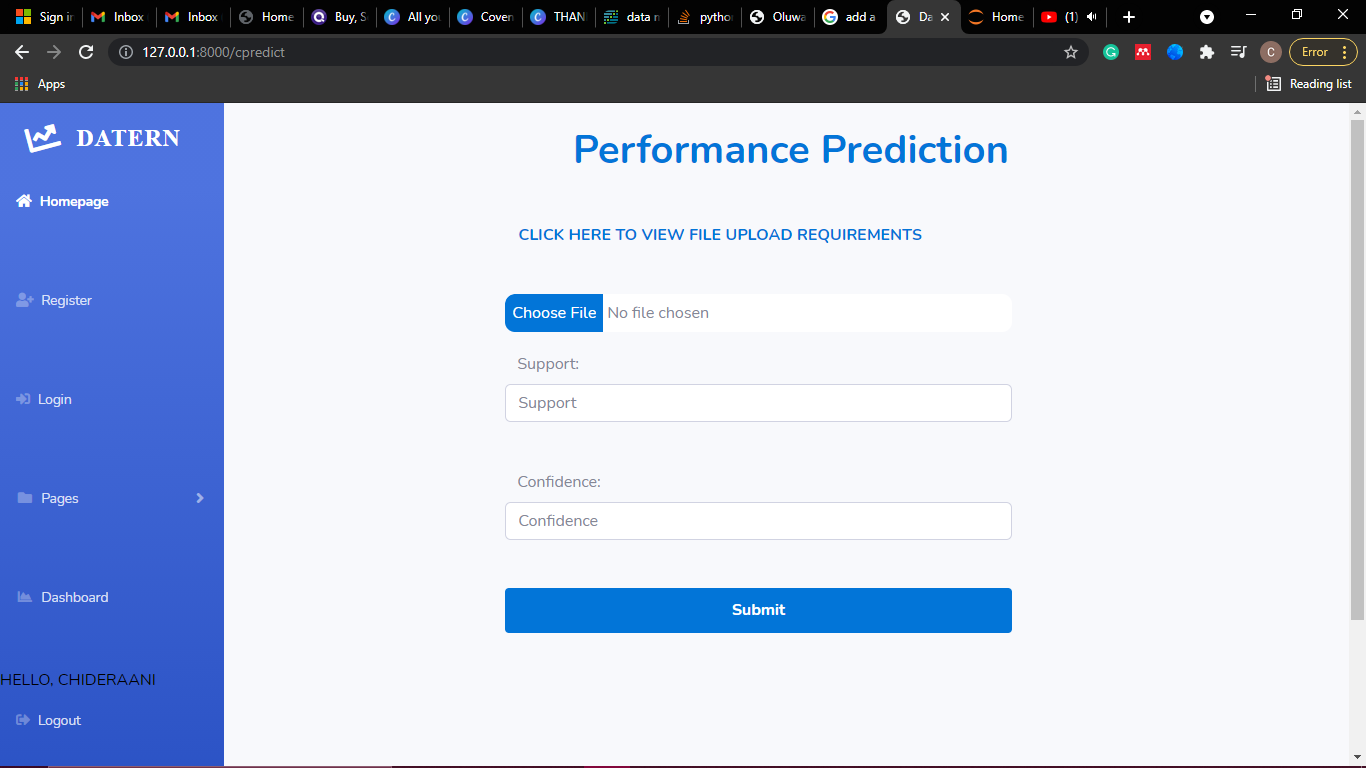
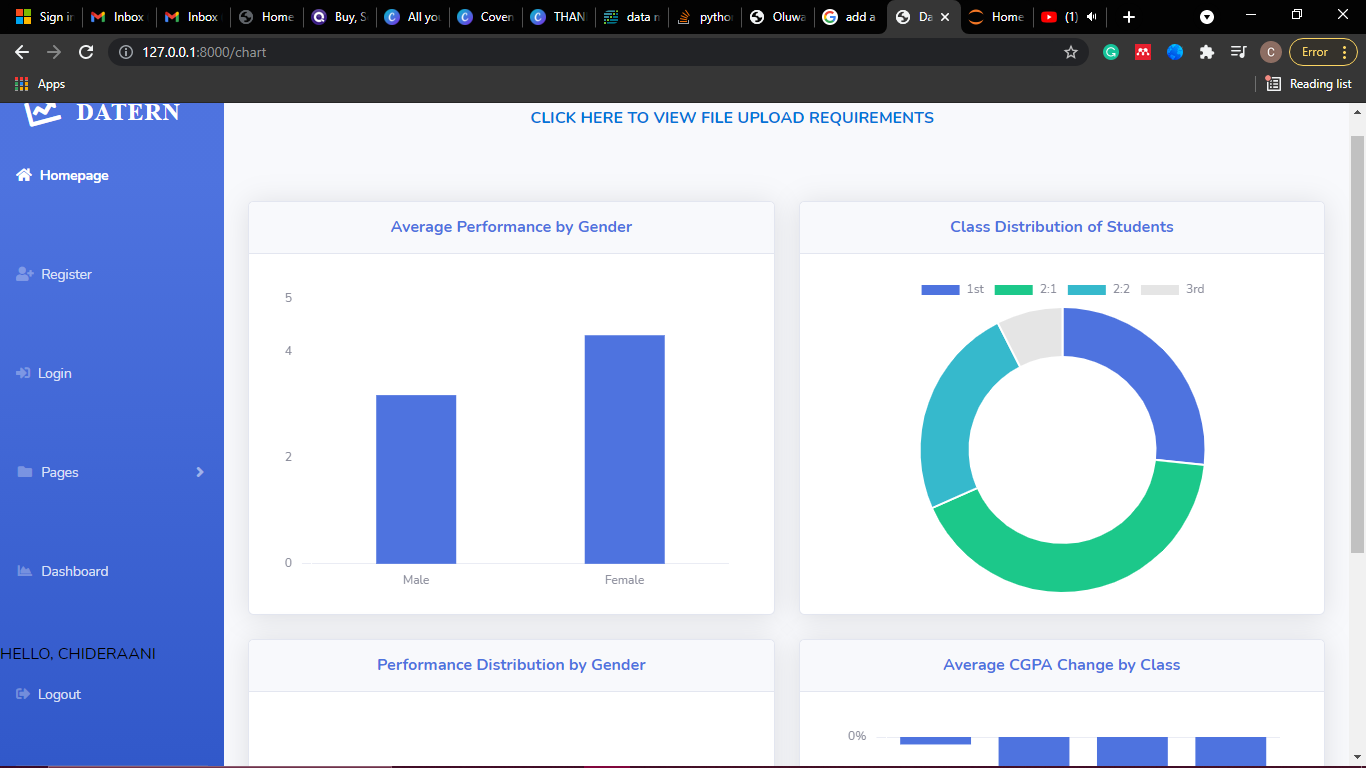
****

Figure 4.38 Course Prediction

### Dashboard

The dashboard page shows the visualisation charts that are generated from a user uploaded file. The dashboards are interactive for maximum user experience. Figure 4.39 shows the dashboard page.



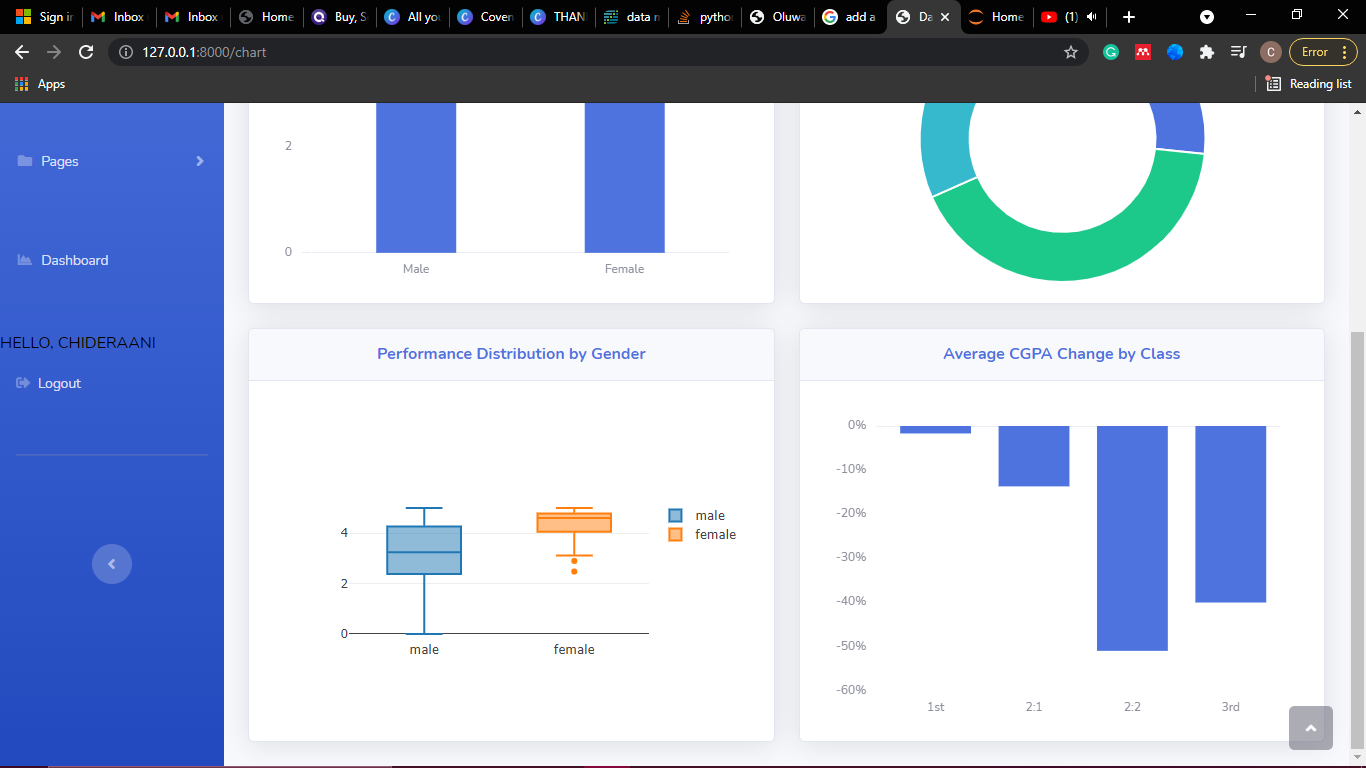


Figure 4.39 Dashboard Page

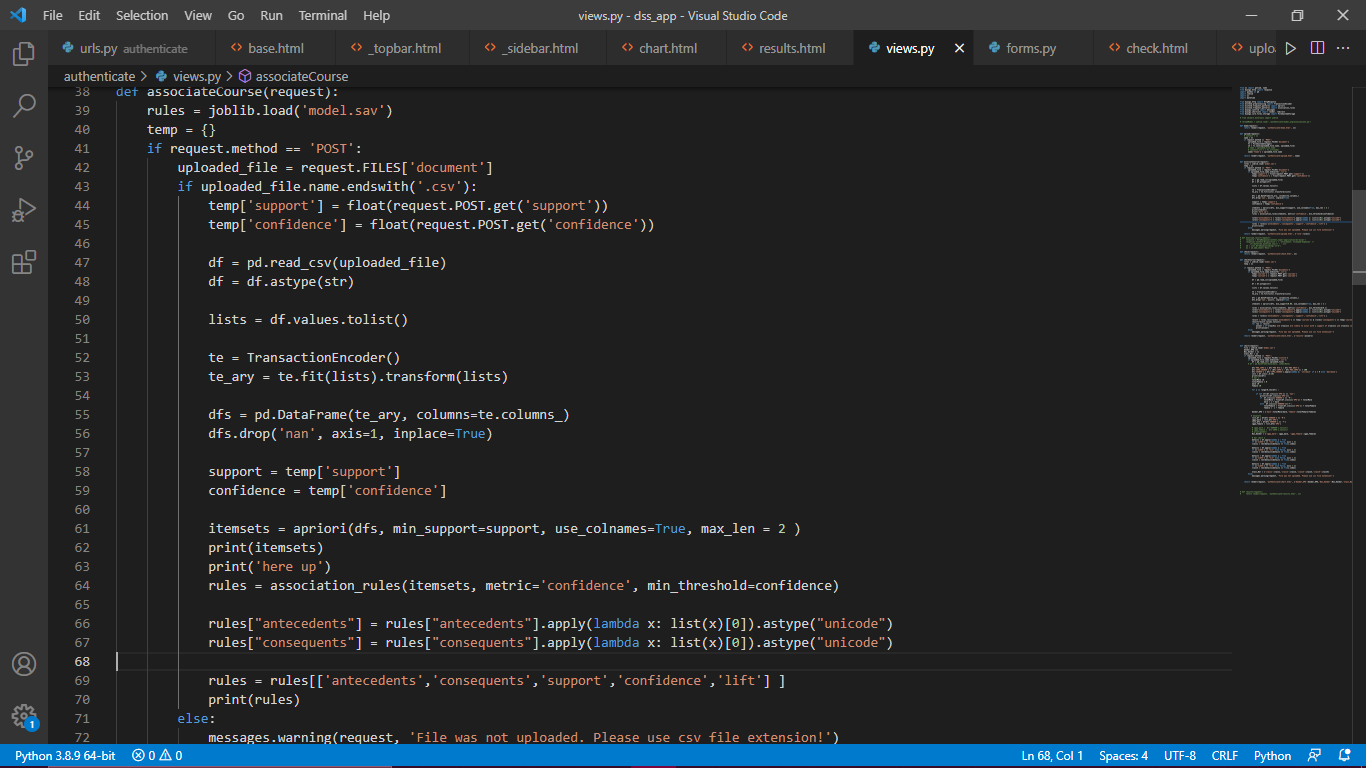


Figure 4.40 Loading model into Django web app

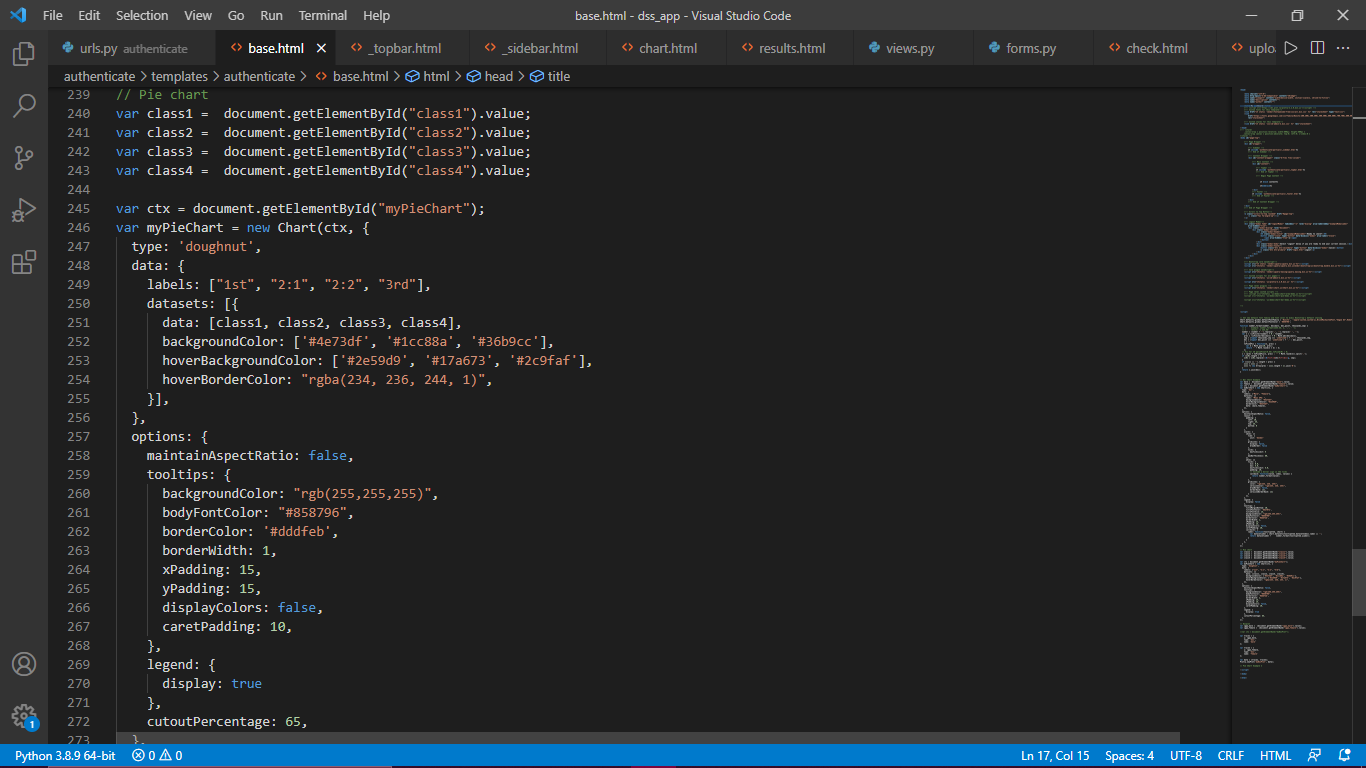


Figure 4.41 Generating charts using chartjs

# CHAPTER FIVE

# SUMMARY, RECOMMENDATION, AND CONCLUSION



## SUMMARY

The primary aim of this study was to investigate numerous patterns in students' academic performance and create an application that would allow users to identify these patterns quickly and easily.

Various association rules were defined from student records using the renowned Apriori method. These guidelines revealed correlations between students' grades in multiple courses. It was also possible to demonstrate how other course structure variables may impact performance. The data determined that a modest curriculum, practical course, and simple course structure resulted in a good performance.

For people to interact with the model, a website application was designed.  Users may upload their data, enter variables, and examine the resulting rules using this system. Hence, the Apriori algorithm has shown to be a reliable approach for identifying patterns and correlations in data. The data gathered from this model and system may be used to make administrative decisions to help students perform better.

## RECOMMENDATION

Understanding which variables influence a student's performance could help create a more suitable structural environment to achieve higher. Some of these elements have been revealed in this study, but further work on this project is still needed to overcome its drawbacks.

Using more data from prior academic years would enhance the accuracy of the results produced from the predictions, which were based on data from only one academic session in this study. Other elements that may influence performance include age, family history, and other psychographic factors (e.g., opinion, interests, extra-curricular activities). Finally, the system only enables users to upload files that are in the CSV format. Other formats, such as 'xlsx' and 'pdf,' might be offered for variety.

## CONCLUSION

Education data mining has made it feasible to analyze and assess a wide range of educational data to enhance learning results. This sector, which some refer to as emergent, has expanded substantially over the years and continues to evolve as researchers continue to look for ways to enhance all areas.

This research initiative has made a significant contribution to this field by examining some elements that influence student success. Rather than taking the conventional approach of forecasting students' future CGPA, this research investigated aspects such as class capacity, which are often neglected. Users of the developed system will be able to establish desired rules and use them to take administrative action and systematically monitor students' performance.

# REFERENCES

Abaidullah, A. M., Ahmed, N., & Ali, E. (2015). *Identifying Hidden Patterns in Students ‟ Feedback through Cluster Analysis*. *7*(1), 16–20. https://doi.org/10.7763/IJCTE.2015.V7.923

Abu-Oda, G. S., & El-Halees, A. M. (2015). DATA MINING IN HIGHER EDUCATION : UNIVERSITY STUDENT DROPOUT CASE STUDY. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, *5*(1), 827–828. https://doi.org/10.2507/daaam.scibook.2009.11

Abu Zohair, L. M. (2019). Prediction of Student’s performance by modelling small dataset size. *International Journal of Educational Technology in Higher Education*, *16*(1), 16-27. https://doi.org/10.1186/s41239-019-0160-3

Aburrous, M., Hossain, M. A., Dahal, K., & Thabtah, F. (2010). Intelligent phishing detection system for e-banking using fuzzy data mining. *Expert Systems with Applications*, *37*(12), 7913–7921. https://doi.org/10.1016/j.eswa.2010.04.044

Admasu, E., & Teklay, A. (2019). Student Performance Prediction with Optimum Multilabel Ensemble Model. *ArXiv*, 1–17.

Aflori, C., & Craus, M. (2007). Grid implementation of the Apriori algorithm. *Advances in Engineering Software*, *38*(5), 295–300. https://doi.org/10.1016/j.advengsoft.2006.08.011

Al-Maolegi, M., & Arkok, B. (2014). AN IMPROVED APRIORI ALGORITHM FOR ASSOCIATION RULES. *International Journal on Natural Language Computing (IJNLC)*, *3*(1), 75-89. https://doi.org/10.5121/ijnlc.2014.3103

Altabrawee, H., Ali, O. A. J., & Ajmi, S. Q. (2019). Predicting Students’ Performance Using Machine Learning Techniques. *JOURNAL OF UNIVERSITY OF BABYLON for Pure and Applied Sciences*, *27*(1), 194–205. https://doi.org/10.29196/jubpas.v27i1.2108

*Apriori Algorithm in Data Mining: Implementation With Examples*. (2011). Retrieved July 2, 2021, from https://www.softwaretestinghelp.com/apriori-algorithm/

Baker, R. S. J. D., & Yacef, K. (2009). The State of Educational Data Mining in 2009 : A Review and Future Visions. *Journal of Educational Data Mining*, *1*(1), 3–16.

Banswal, R., & Madaan, V. (2016). SPACS: Students’ Performance Analysis and Counseling System using Fuzzy logic and Association Rule Mining. *International Journal of Computer Applications*, *134*(3), 12–17. https://doi.org/10.5120/ijca2016907857

Baradwaj, B. K., & Saurabh, P. (2011). *Mining Educational Data to Analyze Students ‟ Performance*. *2*(6), 63–69.

Bin Mohamad, I., & Usman, D. (2013). Standardization and Its Effects on K-Means Clustering Algorithm. *Research Journal of Applied Sciences, Engineering and Technology*. https://www.researchgate.net/profile/Dauda-Usman/publication/288044597\_Standardization\_and\_Its\_Effects\_on\_K-Means\_Clustering\_Algorithm/links/56b5f9b908aebbde1a79bce7/Standardization-and-Its-Effects-on-K-Means-Clustering-Algorithm.pdf

Cantabella, M., Martínez-españa, R., Ayuso, B., & Yáñez, J. A. (2018). *Analysis of student behavior in learning management systems through a Big Data framework*. *90*, 262–272.

Carneiro, N., Figueira, G., & Costa, M. (2017). A data mining based system for credit-card fraud detection in e-tail. *Decision Support Systems*, *95*, 91–101. https://doi.org/10.1016/j.dss.2017.01.002

Caruth, G. D. (2018). Student Engagement, Retention, and Motivation: Assessing Academic Success in Today’s College Students. *Participatory Educational Research*, *5*(1), 17–30. https://doi.org/10.17275/per.18.4.5.1

Castro, F., Vellido, A., Nebot, À., & Mugica, F. (2007). Applying data mining techniques to e-learning problems. *Studies in Computational Intelligence*, *62*(2007), 183–221. https://doi.org/10.1007/978-3-540-71974-8\_8

Cerezo, R., Sánchez-Santillán, M., Paule-Ruiz, M. P., & Núñez, J. C. (2016). Students’ LMS interaction patterns and their relationship with achievement: A case study in higher education. *Computers and Education*, *96*, 42–54. https://doi.org/10.1016/j.compedu.2016.02.006

Chandola, V., & Kumar, V. (2007). *Summarization-Compressing Data into an Informative Representation*.

Chen, F., & Cui, Y. (2020). Utilizing student time series behaviour in learning management systems for early prediction of course performance. *Journal of Learning Analytics*, *7*(2), 1–17. https://doi.org/10.18608/JLA.2020.72.1

Chen, P. H., Yang, F. Y., Lee, D. D., & Yang, M. H. (2018). Data mining the comorbid associations between dementia and various kinds of illnesses using a medicine database. *Computers and Electrical Engineering*, *70*(June), 12–20. https://doi.org/10.1016/j.compeleceng.2018.05.014

Delavari, N., Phon-Amnuaisuk, S., & Beikzadeh, M. R. (2008). Data mining application in higher learning institutions. *Informatics in Education*, *7*(1), 31–54. https://doi.org/10.15388/infedu.2008.03

Dien, T. T., Luu, S. H., Thanh-Hai, N., & Thai-Nghe, N. (2020). Deep learning with data transformation and factor analysis for student performance prediction. *International Journal of Advanced Computer Science and Applications*, *11*(8), 711–721. https://doi.org/10.14569/IJACSA.2020.0110886

Fernandes, E., Holanda, M., Victorino, M., Borges, V., Carvalho, R., & Erven, G. Van. (2019). Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil. *Journal of Business Research*, *94*(August 2017), 335–343. https://doi.org/10.1016/j.jbusres.2018.02.012

Fu, Y. (1997). Data mining. *IEEE Potentials*, *16*(4), 18–20. https://doi.org/10.1109/45.624335

Gatsheni, B. N., & Katambwa, O. N. (2018). *The Design of Predictive Model for the Academic Performance of Students at University Based on Machine Learning*. *6*, 229–237. https://doi.org/10.17265/2328-2223/2018.04.006

Goga, M., Kuyoro, S., & Goga, N. (2015). A Recommender for Improving the Student Academic Performance. *Procedia - Social and Behavioral Sciences*, *180*(November 2014), 1481–1488. https://doi.org/10.1016/j.sbspro.2015.02.296

Hujer, T. (2011). Design and Development of a Compound DSS for Laboratory Research. In *Efficient Decision Support Systems - Practice and Challenges From Current to Future*. InTech. https://doi.org/10.5772/16720

Hussain, M., Zhu, W., Zhang, W., & Abidi, S. M. R. (2018). Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores. *Computational Intelligence and Neuroscience*, *2018*. https://doi.org/10.1155/2018/6347186

Ibrahim, Z., & Rusli, D. (2007). Predicting Students’ Academic Performance: Comparing Artificial Neural Network, Decision tree And Linear Regression. *Proceedings of the 21st Annual SAS Malaysia Forum*, *September*, 1–6. https://www.researchgate.net/profile/Daliela\_Rusli/publication/228894873\_Predicting\_Students’\_Academic\_Performance\_Comparing\_Artificial\_Neural\_Network\_Decision\_Tree\_and\_Linear\_Regression/links/0deec51bb04e76ed93000000.pdf

Jiang, H., Kwong, C. K., Okudan Kremer, G. E., & Park, W. Y. (2019). Dynamic modelling of customer preferences for product design using DENFIS and opinion mining. *Advanced Engineering Informatics*, *42*(October 2018), 100969. https://doi.org/10.1016/j.aei.2019.100969

Kamber, M., Pei, J., & Han, J. (2012). *Data Mining : Concepts and Techniques Third Edition*.

Kasthuriarachchi, K. T. S., & Liyanage, S. R. (2019). Predicting Students’ Academic Performance Using Utility Based Educational Data Mining. *Lecture Notes in Electrical Engineering*, *542*(February), 29–39. https://doi.org/10.1007/978-981-13-3648-5\_4

Kaur, H. (2015). EDM: A Review of Applications of Data Mining in the Field of Education. *India*, *4*(4), 409–412. https://doi.org/10.17148/IJARCCE.2015.4492

Kaur, P., Singh, M., & Josan, G. S. (2015). Classification and Prediction Based Data Mining Algorithms to Predict Slow Learners in Education Sector. *Procedia Computer Science*, *57*, 500–508. https://doi.org/10.1016/j.procs.2015.07.372

Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, *1*(6). https://www.researchgate.net/profile/Trupti-Kodinariya/publication/313554124\_Review\_on\_Determining\_of\_Cluster\_in\_K-means\_Clustering/links/5789fda408ae59aa667931d2/Review-on-Determining-of-Cluster-in-K-means-Clustering.pdf

Lara, J. A., Lizcano, D., Martínez, M. A., Pazos, J., & Riera, T. (2014). A system for knowledge discovery in e-learning environments within the European Higher Education Area - Application to student data from Open University of Madrid, UDIMA. *Computers and Education*, *72*, 23–36. https://doi.org/10.1016/j.compedu.2013.10.009

Lee, D., Park, H., Jung, W., & Chae, Y. (2020). Integrative Medicine Research Identification of candidate medicinal herbs for skincare via data mining of the classic Donguibogam text on Korean medicine. *Integrative Medicine Research*, *9*(4), 100436. https://doi.org/10.1016/j.imr.2020.100436

Liébana-cabanillas, F., Nogueras, R., Herrera, L. J., & Guillén, A. (2013). *Expert Systems with Applications Analysing user trust in electronic banking using data mining methods*. *40*, 5439–5447. https://doi.org/10.1016/j.eswa.2013.03.010

Mahboob, K., Ali, S. A., & Laila, U. e. (2020). Investigating learning outcomes in engineering education with data mining. *Computer Applications in Engineering Education*, *28*(6), 1652–1670. https://doi.org/10.1002/cae.22345

Nagpal, T., & Mishra, M. (2021). Analyzing Human Resource Practices For Decision Making in Banking Sector using HR analytics. *Materials Today: Proceedings*, *xxxx*. https://doi.org/10.1016/j.matpr.2020.12.460

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559–569. https://doi.org/10.1016/j.dss.2010.08.006

Popoola, S. I., Atayero, A. A., Badejo, J. A., John, T. M., Odukoya, J. A., & Omole, D. (2018). *Learning Analytics for smart campus: Data on academic performances of engineering undergraduates in Nigerian private university*.

Prabha, S. L., & Shanavas, A. R. M. (2014). Educational Data Mining Applications. *Operations Research and Applications: An International Journal (ORAJ)*, *1*(1), 23–29.

Quadri, M., & Kalyankar, D. (2010). Drop out feature of student data for academic performance using decision tree techniques. *Global Journal of Computer*, *10*(2), 2–5. http://computerresearch.org/stpr/index.php/gjcst/article/viewArticle/128

Ramani, R. G., & Sivagami, G. (2011). Parkinson Disease classification using data mining algorithms. *International Journal of Computer Applications*, *32*(9), 17–22.

Ramaswami, M., & Bhaskaran, R. (2010). *A CHAID Based Performance Prediction Model in Educational Data Mining*. *7*(1), 10–18. http://arxiv.org/abs/1002.1144

Rastrollo-Guerrero, J. L., Gómez-Pulido, J. A., & Durán-Domínguez, A. (2020). Analyzing and predicting students’ performance by means of machine learning: A review. *Applied Sciences (Switzerland)*, *10*(3). https://doi.org/10.3390/app10031042

Romero, Cristóbal, & Ventura, S. (2010). Educational Data Mining: A Review of the State of the Art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*.

Romero, Cristobal, Ventura, S., Pechenizkiy, M., & Baker, R. S. J. D. (2010). *Handbook of educational data mining*. CRC Press. https://books.google.com.ng/books?hl=en&lr=&id=u5aWVw0uQJMC&oi=fnd&pg=PP1&dq=Handbook+of+educational+data+mining&ots=KD-QCH068M&sig=ZDKuQZZ9Abl\_KEMF5O94Cxr-zZ4&redir\_esc=y#v=onepage&q=Handbook of educational data mining&f=false

Sandoval, A., Gonzalez, C., Alarcon, R., Pichara, K., & Montenegro, M. (2018). Centralized student performance prediction in large courses based on low-cost variables in an institutional context. *Internet and Higher Education*, *37*(February), 76–89. https://doi.org/10.1016/j.iheduc.2018.02.002

Şen, B., Uçar, E., & Delen, D. (2012). Predicting and analyzing secondary education placement-test scores: A data mining approach. *Expert Systems with Applications*, *39*(10), 9468–9476. https://doi.org/10.1016/j.eswa.2012.02.112

Seufert Sabine, Christoph Meier, Matthias Soellner, R. R. (2019). *A Pedagogical Perspective on Big Data and Learning Analytics: A Conceptual Model for Digital Learning Support*.

Shahiri, A. M., Husain, W., & Rashid, N. A. (2015). A Review on Predicting Student’s Performance Using Data Mining Techniques. *Procedia Computer Science*, *72*, 414–422. https://doi.org/10.1016/j.procs.2015.12.157

Tomasevic, N., Gvozdenovic, N., & Vranes, S. (2020). An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers and Education*, *143*(February 2019), 103676. https://doi.org/10.1016/j.compedu.2019.103676

Ünal, F. (2020). Data Mining for Student Performance Prediction in Education. *IntechOpen*.

Varela, N., Valega, J. A. G., & Lezama, O. B. P. (2020). Analysis of behavior of automatic learning algorithms to identify criminal messages. In *Procedia Computer Science* (Vol. 175, pp. 114–119). https://doi.org/10.1016/j.procs.2020.07.019

Wei, J. T., Lee, M. C., Chen, H. K., & Wu, H. H. (2013). Customer relationship management in the hairdressing industry: An application of data mining techniques. *Expert Systems with Applications*, *40*(18), 7513–7518. https://doi.org/10.1016/j.eswa.2013.07.053

Yoon, B., Jeong, Y., Lee, K., & Lee, S. (2020). A systematic approach to prioritizing R&D projects based on customer-perceived value using opinion mining. *Technovation*, *98*(August), 102164. https://doi.org/10.1016/j.technovation.2020.102164

Zeineddine, H., Braendle, U., & Farah, A. (2021). Enhancing prediction of student success: Automated machine learning approach. *Computers and Electrical Engineering*, *89*(October 2019), 106903. https://doi.org/10.1016/j.compeleceng.2020.106903

# APPENDIX

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pylab as plt

df = pd.read\_csv('C:/Users/CHIDERA ANI/Downloads/project stuff/data/gender/CSC 200LVL.csv')

df.drop(['S/N','NAME'], axis=1,inplace=True)

df.head()

columns = ['GENDER', 'AGE','N FAIL','C FAIL','FAILED COURSES','INCOMPLETE COURSES','TOT WTS','TOT UNIT','PRE WTS','PRE UNIT','CUMWTS',\

'CUMUNIT','CGPA','STAND','CLASS','TMC 212','PHY 119', 'ACC 111', 'BIO 112']

dfc = dfc.drop(columns, axis=1)

dfc.fillna(0, inplace=True)

dfc.dropna(axis=1,how='all',inplace=True)

dfc.head()

sns.displot(df['CGPA'], bins=10)

plt.rcParams["figure.figsize"] = fig\_size

sns.distplot(df['CGPA'], bins=10)

plt.rcParams["figure.figsize"] = (20, 18)

df.hist()

plt.savefig('histplot')

f, ax = plt.subplots(figsize=(20,18))

mask = np.triu(np.ones\_like(dfc.corr(), dtype=bool))

sns.heatmap(dfc.corr(), mask=mask, vmax=.3, center=0, annot=True,

square=True, linewidths=.5, cbar\_kws={"shrink": .8})

# sns.heatmap(dfc.corr(), annot=True)

f, ax = plt.subplots(figsize=(8,6))

df['GENDER'].value\_counts(normalize=True).plot(kind='barh')

df['PRE\_CGPA'] = df['PRE WTS'] / df['PRE UNIT']

df['CGPA\_CHANGE'] = (df['CGPA'] - df['PRE\_CGPA']) \* 100

df['CHANGE'] = df['CGPA\_CHANGE'].apply(lambda x: 'increase' if x > 0 else 'decrease')

df.head(2)

# average performance by gender

from numpy import mean

g= sns.barplot(x='GENDER', y='GPA', data=dfs,estimator=mean, ci=None)

# plt.rcParams["figure.figsize"] = (10, 15)

plt.title('Average Performance by Gender', size=20)

plt.xlabel('Gender', size=15)

plt.ylabel('Average GPA', size=15)

for p in g.patches:

g.annotate(format(p.get\_height(), '.3f'), (p.get\_x() + p.get\_width() / 2., p.get\_height()), ha = 'center', \

va = 'center', xytext = (0, 10), textcoords = 'offset points')

# average performance by class and gender

from numpy import mean

g= sns.barplot(x='CLASS', y='GPA', hue='GENDER',data=dfs,estimator=mean, ci=None)

# plt.rcParams["figure.figsize"] = (10, 15)

plt.title('Average Performance by Gender & Class', size=20)

plt.xlabel('Gender', size=15)

plt.ylabel('Average GPA', size=15)

for p in g.patches:

g.annotate(format(p.get\_height(), '.3f'), (p.get\_x() + p.get\_width() / 2., p.get\_height()), ha = 'center', \

va = 'center', xytext = (0, 10), textcoords = 'offset points')

# boxplot distribution of gender&gpa

sns.boxplot(x='GENDER', y='CGPA', data=dfs)

df.fillna(0, inplace=True)

df.isnull().sum()

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

norm\_df = scaler.fit\_transform(df.drop(columns=['GENDER']))

norm\_df = pd.DataFrame(norm\_df, columns=df.drop(columns=['GENDER']).columns)

norm\_df['GENDER'] = df['GENDER']

columns = df.iloc[:,0]

norm\_df.drop('GENDER', axis=1).astype('float')

# class distribution of students

g = sns.countplot(x='CLASS',data=df, color='blue')

plt.title('Class Distribution of Students')

plt.xlabel('Class')

plt.ylabel('Count')

plt.rcParams["figure.figsize"] = (10, 8)

for p in g.patches:

g.annotate(format(p.get\_height()), (p.get\_x() + p.get\_width() / 2., p.get\_height()), ha = 'center', \

va = 'center', xytext = (0, 10), textcoords = 'offset points')

# percentage change in cgpa

dataForPlot = df.groupby('CLASS').mean().CGPA\_CHANGE

fig, ax = plt.subplots()

ax.bar(dataForPlot.index, dataForPlot, color=['blue'])

# ax.set\_xticks([0, 1], False)

ax.set\_xlabel('Class')

ax.set\_ylabel('Average CGPA Change')

ax.set\_title('Percentage Change in CGPA by Class')

plt.rcParams["figure.figsize"] = (10, 8)

for p in ax.patches:

ax.annotate(format(p.get\_height(), '.2f'), (p.get\_x() + p.get\_width() / 2., p.get\_height()), ha = 'center', \

va = 'center', xytext = (0, 10), textcoords = 'offset points')

# previous vs current cgpa

plt.rcParams["figure.figsize"] = (10, 8)

sns.scatterplot(x='CGPA', y='PRE\_CGPA', hue='CHANGE', data=df, s=90)

plt.title('CGPA VS PREVIOUS CGPA SHOWING INCREASE/DECREASE', size=20)

plt.xlabel('CURRENT CGPA', size=15)

plt.ylabel('PREVIOUS CGPA', size=15)

# boxplot distribution of courses

plt.rcParams["figure.figsize"] = (20, 10)

sns.boxplot(x="variable", y="value", data=pd.melt(cols))

plt.show()

dfc.drop('GPA',axis=1,inplace=True)

# sclaing the variables for clustering

from sklearn.preprocessing import StandardScaler

X = dfc.iloc[:,:]

scaler = StandardScaler()

norm\_df = scaler.fit\_transform(X)

# elbow method to determine K (number of clusters)

from sklearn.cluster import KMeans

kmeans\_kwargs = {

"init": "random",

"n\_init": 10,

"max\_iter": 300,

"random\_state": 0,

}

# elbow method

cluster\_range=range(1,11)

sse = []

for k in cluster\_range:

kmeans = KMeans(n\_clusters=k, \*\*kmeans\_kwargs)

kmeans.fit(norm\_df)

sse.append(kmeans.inertia\_)

cluster\_df = pd.DataFrame({'num\_clusters':cluster\_range,'sse':sse})

cluster\_df[0:10]

plt.rcParams["figure.figsize"] = (8, 6)

plt.plot(cluster\_df.num\_clusters, cluster\_df.sse, marker='+')

plt.xlabel('Number of Clusters')

plt.ylabel('SSE')

from kneed import KneeLocator

kl = KneeLocator(

range(1, 11), sse, curve="convex", direction="decreasing"

)

kl.elbow

#using silhouette coefficient

from sklearn.metrics import silhouette\_score

silhouette\_coefficients = []

cluster\_range = range(2, 11)

for k in cluster\_range:

kmeans = KMeans(n\_clusters=k, \*\*kmeans\_kwargs)

kmeans.fit(norm\_df)

score = silhouette\_score(norm\_df, kmeans.labels\_)

silhouette\_coefficients.append(score)

silh\_df = pd.DataFrame({'num\_clusters':cluster\_range,'silhouette\_coefficients':silhouette\_coefficients})

silh\_df[0:10]

plt.rcParams["figure.figsize"] = (8, 6)

plt.style.use("fivethirtyeight")

plt.plot(range(2, 11), silhouette\_coefficients)

plt.xticks(range(2, 11))

plt.xlabel("Number of Clusters")

plt.ylabel("Silhouette Coefficient")

plt.show()

# using gradient descent

from sklearn.metrics import pairwise\_distances

def compute\_inertia(a, X):

W = [np.mean(pairwise\_distances(X[a == c, :])) for c in np.unique(a)]

return np.mean(W)

def compute\_gap(clustering, data, k\_max=5, n\_references=5):

if len(data.shape) == 1:

data = data.reshape(-1, 1)

reference = np.random.rand(\*data.shape)

reference\_inertia = []

for k in range(1, k\_max+1):

local\_inertia = []

for \_ in range(n\_references):

clustering.n\_clusters = k

assignments = clustering.fit\_predict(reference)

local\_inertia.append(compute\_inertia(assignments, reference))

reference\_inertia.append(np.mean(local\_inertia))

ondata\_inertia = []

for k in range(1, k\_max+1):

clustering.n\_clusters = k

assignments = clustering.fit\_predict(data)

ondata\_inertia.append(compute\_inertia(assignments, data))

gap = np.log(reference\_inertia)-np.log(ondata\_inertia)

return gap, np.log(reference\_inertia), np.log(ondata\_inertia)

k\_max = 5

gap, reference\_inertia, ondata\_inertia = compute\_gap(KMeans(), norm\_df, k\_max)

plt.plot(range(1, k\_max+1), reference\_inertia,

'-o', label='reference')

plt.plot(range(1, k\_max+1), ondata\_inertia,

'-o', label='data')

plt.xlabel('k')

plt.ylabel('log(inertia)')

plt.show()

plt.plot(range(1, k\_max+1), gap, '-o')

plt.ylabel('gap')

plt.xlabel('k')

# plotting clusters

kmeans = KMeans(n\_clusters=3, init='k-means++', max\_iter=300, n\_init=10, random\_state=0)

pred\_y = kmeans.fit\_predict(norm\_df)

plt.scatter(norm\_df[pred\_y == 0, 0], norm\_df[pred\_y == 0, 1], s = 100, c = 'red', label = 'Cluster 1')

plt.scatter(norm\_df[pred\_y == 1, 0], norm\_df[pred\_y == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')

plt.scatter(norm\_df[pred\_y == 2, 0], norm\_df[pred\_y == 2, 1], s = 100, c = 'green', label = 'Cluster 3')

plt.legend()

plt.show()

norm\_df = pd.DataFrame(norm\_df, columns=dfc.columns, index=dfc.index)

norm\_df.head(2)

centroids = pd.DataFrame(kmeans.cluster\_centers\_,columns=norm\_df.columns)

pd.set\_option('precision', 3)

centroids

distances = kmeans.transform(norm\_df)

minSquaredDistances = distances.min(axis=1) \*\* 2

data = {'squaredDistance': minSquaredDistances,'cluster': kmeans.labels\_}

df\_plot = pd.DataFrame(data, index=norm\_df.index)

for cluster, data in df\_plot.groupby('cluster'):

count = len(data)

withinClustSS = data.squaredDistance.sum()

print(f'Cluster {cluster} ({count} members): {withinClustSS:.2f} within cluster')

from sklearn.metrics import pairwise

pd.DataFrame(pairwise.pairwise\_distances(kmeans.cluster\_centers\_, metric='euclidean'))

norm\_df['cluster'] = kmeans.labels\_

norm\_df.head(2)

# profiling chart

columns = norm\_df.drop('cluster', axis=1).columns

df\_nor\_melt = pd.melt(norm\_df,

id\_vars=['cluster'],

value\_vars=columns,

var\_name='Attribute',

value\_name='Value')

df\_nor\_melt.head()

sns.set(rc={'figure.figsize':(20,10)})

sns.lineplot('Attribute', 'Value', hue='cluster', data=df\_nor\_melt,palette=['lime','tomato','blue'],legend='full')

# association mining 1

import pandas as pd

import numpy as np

df = pd.read\_csv('C:/Users/CHIDERA ANI/Downloads/project stuff/data/300level\_MIS\_assoc.csv')

df.head()

df = df.astype(str)

lists = df.values.tolist()

lists

from mlxtend.preprocessing import TransactionEncoder

te = TransactionEncoder()

te\_ary = te.fit(lists).transform(lists)

te\_ary

dfs = pd.DataFrame(te\_ary, columns=te.columns\_)

dfs.drop('nan', axis=1, inplace=True)

dfs.head()

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

# create frequent itemsets

itemsets = apriori(dfs, min\_support=0.05, use\_colnames=True, max\_len = 2 )

itemsets

# convert into rules

rules = association\_rules(itemsets, metric='confidence', min\_threshold=0.1)

rules.head()

rules1 = rules[['antecedents','consequents','support','confidence','lift']]

rules1 = rules1.sort\_values(['confidence'], ascending =[False])

rules1.head()

rules1["antecedents"] = rules1["antecedents"].apply(lambda x: list(x)[0]).astype("unicode")

rules1["consequents"] = rules1["consequents"].apply(lambda x: list(x)[0]).astype("unicode")

result = rules1.loc[(rules1['antecedents'] == 'BUS313\_C') & (rules1['consequents'] == 'BFN311\_B')]

result=result.values.tolist()

# association 2

import pandas as pd

import numpy as np

df = pd.read\_csv('C:/Users/CHIDERA ANI/Downloads/project stuff/data/association1.csv')

df.set\_index('S/N', inplace=True)

df.head()

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

itemsets = apriori(df, min\_support=0.2, use\_colnames=True)

itemsets.head()

# convert into rules

rules = association\_rules(itemsets, metric='confidence', min\_threshold=0.6)

rules.head()

rules["antecedents"] = rules["antecedents"].apply(lambda x: list(x)[0]).astype("unicode")

rules["consequents"] = rules["consequents"].apply(lambda x: list(x)[0]).astype("unicode")

rules.head()

rules = rules[['antecedents','consequents','support','confidence','lift']]

rules = rules.sort\_values(['confidence'], ascending =[False])

rules.head()

result\_perf = rules.loc[(rules['consequents'] == 'CPerformance\_Poor')| (rules['consequents'] == 'CPerformance\_Fair') | \

(rules['consequents'] == 'CPerformance\_Good')]

result\_perf.head(3)