Pamantasan ng Lungsod ng Maynila

**AN INTELLIGENT STUDENT ADVISING SYSTEM:
A MINIMUM SPANNING TREE APPROACH**

A Capstone Project Presented to the Graduate Program
College of Engineering and Technology
Pamantasan ng Lungsod ng Maynila

In Partial Fulfillment of the Requirements for the Degree
Master in Information Technology

By
Vivien A. Agustin

Dr. Khatalyn E. Mata
Thesis Adviser

March 2021

APPROVAL SHEET

The capstone project hereto titled

**AN INTELLIGENT STUDENT ADVISING SYSTEM:
A MINIMUM SPANNING TREE APPROACH**

prepared and submitted by Vivien A. Agustin in partial fulfilment of the requirements for the degree of Master in Information Technology has been examined and is recommended for acceptance and approval for **ORAL EXAMINATION**.

____

DR. KHATALYN E. MATA
Adviser

PANEL OF EXAMINERS

Approved by the Committee on Oral Examination
with a grade of __________ on ________.

PROF. MANUEL L. OCAMPO
Panel Chair
Chairman

PROF. EDGARDO S. DAJAO CORTEZ
Panel Member
Member

PROF. DAN MICHAEL A.
Panel Member
Member

Accepted and approved in partial fulfilment of the requirements for the degree of Master in Information Technology

____

____

DR. JOSEPH BERLIN P. JUANZON

ENGR. JUAN C. TALLARA JR.

Director, CET
Graduate Program

Dean
Graduate Program

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER One

INTRODUCTION

1.1 Background of the Study

For the past years, the use of information to ease the struggles of administrators, teachers, and students has been the focus of education, researchers, and theories. Efficient data management in schools, just like any organization, has always been crucial in the attainment of its goals and objectives. Forman (2007), in his article "New Research Perspective on Mobility, Organizations, Systems, and Technologies", countless educational institutions have opted to adopt technological innovations to improve their daily operations and easier accessibility of information among the end-users. Online registration systems, for instance, allow students to register through the internet, eliminating or reducing problems with traditional registration systems, such as long queues, paper forms, and troublesome waitlists.

The Universidad De Manila, being a government-funded university, deals with more than ten thousand students every enrollment, which is expected to grow, especially now that its programs and services are gradually disclosed to the public. At present, Universidad de Manila is utilizing an Online-based enrollment system. Although this is a great leap from the tedious manual process, it still has limitations that the researcher could address. The Online-based Enrollment System's capability is somehow limited to just the evaluation of student's grades and registration.

Part of the enrollment process is the academic subject advising which has always been complicated and time-consuming especially with the increasing number of enrolling students. Unfortunately, the current system being utilized by the university does not have this capability. The procedure requires scrutiny of the student's progress or performance throughout his college education and the faculty members are tasked to advise the student, especially the irregular students on what subjects to take in that particular semester. The faculty members consume at least 5-7 minutes in evaluating and advising irregular students on what subject to enroll. This also varies depending on the case of a student, especially for those with complicated cases. Subject advising is a critical activity for the faculty members as they need to investigate the individual cases of irregular students to determine the appropriate subjects that they need to enroll. The faculty members are also responsible for providing the student with timely and correct information to enable them to fulfill the requirements of their academic degree. With these identified issues, there is a real need to introduce an intelligent subject advising application to lessen the burden of the academic advisors, and assist both the students and the faculty members in accomplishing their tasks, given its ability to deal with large quantities of data.

With that being said, the researcher recommends this study entitled, An Intelligent Student Advising System: A Minimum Spanning Tree Approach, which will utilize machine learning techniques to develop an academic advisory for students. This proposed study will help the students make sound decisions on what subject to take to guide students on the right track using Weighted Minimum Spanning Tree, and at the same time the application will flag students who are at risk of not graduating on time. The application will automate the decision making to help students in their enrollment process. It will optimize the enrollment process and reduce the effort of the faculty-in-charge during enrollment and even the students. This study is ultimately aimed at improving the efficiency of Universidad de Manila through the employment of an intelligent student advising system using machine learning techniques.

1.2 Statement of the Problem

At present, Universidad de Manila is utilizing an Online-based enrollment system. Although this is a great leap from the tedious manual process, it still has limitations that the researcher could address. One of the processes that can be greatly improved is the academic advising. This is where a list of subjects that should be taken by the students is manually done by the faculty-in-charge during enrollment. Grades of students are manually assessed and analyzed by the faculty-in-charge to determine their academic status.

1.3 Objective of the Study

This study aims to develop an intelligent web-based student advising system for Universidad de Manila that will alleviate, if not eliminate the time consuming process in subject advising.

Specifically, the study seeks to address the following objectives:

1. To develop an application that will recommend the list of subjects to be enrolled by the students utilizing the concept behind the minimum spanning tree.
2. To develop an application that will flag students who are at risk of not graduating on time.
3. To develop an application that will determine the academic status of students.
4. To evaluate the newly developed system in terms of accuracy.

1.4 Scope and Limitations

The study will be focusing on the development of a web-based intelligent student advising system for Universidad de Manila using machine learning which is minimum spanning tree. The Minimum Spanning Tree (MST) is utilized to represent the program of study of the students. MST is used for approximating the traveling salesman problem. This problem is similar to how the program of study can be represented. Since each subject on the map is required to be taken by the student, all nodes should be visited. The ultimate goal is to identify the shortest path while visiting all courses. It further becomes more complex with the irregular students. Numerous constraints need to be considered. These include the maximum number of units that can be taken each semester, the lecture and

laboratory hours, and the prerequisite courses. To identify the shortest path, each edge connecting each node should be labeled or assigned by a certain value - the weights.

The proposed application will be able to advise the students as to which subjects should they take for the next semester. The application can also flag students who are at risk of not graduating on time. The system can also evaluate the grades of the students and determine their academic status.

The proposed study excludes the enrollment process, thus, it is more focused on the development of an accurate and time efficient model. The model developed was used only to simulate the advising process. On actual implementation, it can be integrated to the student information system through the retrieval of the stored grades of the student. Including the database pull out of data will require a massive development time, considering the schema and all associated data therein, thus, was included in the recommendation for future work. The study is limited to three courses: the Bachelor of Science in Criminology, Bachelor of Science in Information Technology and Bachelor of Science in Electronics Communication and Engineering, due to the availability of the historical data used in developing the model

1.5 Significance of the Study

Results obtained from this study will benefit the following stakeholders:

Students. The students are the main clients of Universidad de Manila. As such, the proposed application will immensely benefit them in terms of prompt services and accurate records. With the proposed system, the students, especially the irregular students, will be advised with the appropriate subjects that they need to enroll. The proposed application has the ability to recommend a list of subjects to be enrolled by the students and they will also be flagged for the possibility of not graduating on time.

Faculty Members. The proposed application automates the subject to be taken by the students. With this feature, the faculty members need not to report to school just to assist in advising the students on what subject to take for the next semester. It will also help lessen their work and the time consumed in handling this task during enrollment.

Future Researchers. With this study, the researchers will be guided on the conduct of a new research of the same nature. With this study as a template, they can employ other alternative solutions by employing other models or algorithms that best suit the needs of the university.

In summary, the study will ensure quality service in response to the clients' needs, especially the students and faculty members of Universidad de Manila.

Chapter Two

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter presents the different researches and other literatures form both foreign and local researchers, which have significant bearings on the variables included in the research. It focuses on several aspects that will help in the development of this study. The study is generally concentrated on creating a system for the Universidad De Manila. The literatures of this study come from books, journals, articles, electronic materials such as PDF or E-Book, and other existing thesis and dissertations, foreign and local which are believed to be useful in the advancement of awareness concerning the study.

2.1Related Literature

a. Minimum Spanning Tree

A Minimum Spanning Tree is a subset of the edges which connect all vertices in the graph with the minimal total edge cost. The weight of a spanning tree is the number of weights given to each of the spanning tree's edges. In particular, any spanning tree of that graph is minimal if all the edge weights of a given graph are the same. Multiple minimum spanning trees of the same weight can exist. If each edge has a separate weight, only one particular minimum spanning tree is available (F. Shalik, S. Besawada, N. Goveas, 2015). Based on the article authored by M. Beuret, N. Billot, L. Cambresy, D.J. Eden, S. Pezzuto, D. Elia, and E. Schisano, 2017, the minimum spanning tree (MST) method was used to describe the over-densities in two dimensions. By heliocentric distance folding, the list was further refined, resulting in more detailed over-densities that are candidates for clusters

To derive a Minimum Spanning Tree, Prim's algorithm or Kruskal's algorithm can be used. In this study the Kruskal's algorithm will be utilized. The Kruskal's Minimum Spanning Tree algorithm is shown below:

1. Sort all the edges in non-decreasing order of their weight.
2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include its edge. Else, discard it.
3. Repeat step#2 until there are (V-1) edges in the spanning tree.

Table 2.1 Comparison between Kruskal and Prim Algorithm

	Kruskal's Algorithm	**Prim's Algorithm**
Multiple MST's	Offers a good control over the resulting MST	Controlling the MST might be a little harder
Implementation	Easier to implement	Harder to implement
Requirements	Disjoint Set	Priority Queue

Table 2.1 shows that Kruskal's algorithm outperforms other methods in common scenarios because it uses simplified data structures. In sparse graphs, Kruskal's algorithm performs better compared to Prim's algorithm. Kruskal's algorithm starts from the least weighted edge to create the minimum spanning tree, while Prim's algorithm starts from the root vortex. Kruskal's algorithm can create a forest at any time and work with disconnected components while Prim's algorithm returns a connected part and only works for connected graphs. In terms of ease of implementation and power over the resulting MST, the Kruskal algorithm is preferable. Prim's algorithm has a higher level of difficulty while Kruskal's algorithm is more reliable, precise, and sensitive.

b. Ensemble Algorithm

Building a fair model from a dataset is one of the key tasks of machine learning algorithms, based on the journal titled 'Basics of Ensemble Learning in

Classification Techniques Explained' by Ambika Choudhury, 2019. The process of constructing data models is learning or teaching, and the learned model may be referred to as a theory or a learner. By choosing their predictions, the learning algorithms that construct a collection of classifiers and then classify new data points are known as Ensemble methods. In order to minimize risk, ensemble-based machine learning is a decision-making technique. Bagging, boosting, stacked generalization and mixing of expert methods are the most common techniques for constructing ensemble structures. For the purposes of integrating class labels, weighted majority voting, action information space and border count methods are used to combine class label outputs. And accomplish the diversity among the classifiers that is central to ensemble learning (Shahid Ali, Sreenivas Sremath Tirumala, Abdolhossein Sarrafzadeh, 2015).

To obtain better predictive performance than a single classifier of decision trees, ensemble strategies combine multiple classifiers of decision trees. The primary theory behind the ensemble model is that a group of weak learners come together to form a strong learner, thus improving the accuracy of the model (Jinde Shubham, 2018). The ensemble will increase generalization ability and robustness. A significant problem in the direction of research in machine learning is the classification of the ensemble. Another significance of the ensemble is that it is much better than a single base learner to produce correct hypotheses (M. Sravan Kumar Reddy, K.E. Naresh Kumar, and Dharmendra Singh Rajput, 2019). There is a long history of developing machine learning

techniques that solve this problem by learning multiple models and integrating their results instead of only learning a single model. For the ensemble, these approaches are referred to as methods or ensembles. An ensemble is a group of (base) predictive models developed with a
given algorithm that is supposed to lead to predictive output gain over an individual model by combining the predictions of its constituents (Matej Petkovic, Redouane Boumgha, Martin Breskvar, Saso Dzeroski, Dragi Kocev, Jurica Levatic, Luke Lucas, Aljaz Osojnik, Nikola Simidjievski, 2019) .

According to Zhiyuan Ma, Ping Wang, Zehui Gao, Ruobing Wang, and Koroush Khalighi, 2018, the "bagging" ensemble approach was applied to predict the dose of warfarin. Another ensemble approach that uses a higher-level model to combine lower-level models to achieve higher accuracy of prediction is stacked generalization. Stacked generalization can combine different kinds of algorithms, unlike the "bagging" and "boosting" methods that can only combine machine learning algorithms of the same kind. The advantages of Ensemble algorithms were only evident when a large dataset was used in which the base algorithms were outperformed by Stacking and the results obtained support the idea that Ensemble algorithms were able to perform better than individual base algorithms (Eu Jin Phua and Nowskath Kadhar Batcha, 2019). Ensemble methods are a technique in machine learning that combines many basic models to create one optimal predictive model. Ensemble Methods allow us to take into account a sample of Decision Trees, determine the features to use or questions to ask at each section, and make a final predictor based on the sampled Decision Trees' aggregated results (Evan Lutins, 2017). To achieve improved predictive efficiency than could be obtained from any of the constituent learning algorithms alone, machine learning ensemble approaches incorporate many learning algorithms. Theoretically and experimentally, the combination of multiple learning models is shown to provide substantially improved results than their single base learners (Panagiotis Pintelas and Ioannis E. Livieris, 2020).

According to Mónica Villaverde, David Aledo, and David Perez, 2019, the accuracy of a classification can be enhanced by ensemble techniques. Usually, two stages are composed of an ensemble. It generally includes a primary set of learners, also referred to as basic learners, who are in charge of generating second stage estimates, all of which must be combined. Bagging, boosting, stacked generalization, and sets of learners are the most popular ensemble techniques.

c. Random Forest

A supervised learning algorithm which can be used for classification and regression is Random Forest. It is the algorithm that is the most versatile and simplest to use. A forest is made up of trees, and it is said that the more trees it has, the better the forest will be. On randomly selected data samples, Random Forest generates decision trees, gets predictions from each tree and selects the best solution by voting. It has a number of applications, such as the recommendation engine, the classification of images and the option of features. It can also be used to categorize loyal applicants for loans, recognize fraudulent behavior and predict diseases.

In formative years, selecting the right path is a very critical decision since this decision depends on his future. The student alone is not sufficiently mature to make the right decision in his early life. Choosing incorrect courses implies a difference between the skill, ability and personal interest of the student. Since no other credible source is commonly available that can guide the student in the most appropriate way, a recommendation framework has been built to provide him feedback on choosing the right course. The Recommender Framework is a program built with the aid of experts that helps to find a path for their future research in the specifics of the students' history and their skills. The study suggests potential predictions for the selection of courses for students based on their marks and choice of interest in the work.

The technique of clustering is used to classify structures and relationship within the data. Grewal DS and Kaur, 2016 Random Forests are constructed by combining the predictions of many trees, each of which is isolated. The trees are individually trained and the trees' predictions are averaged by averaging. When building a random tree, there are three key choices to be made. These are the way the leaves are separated, the type of predictor to be used in each leaf, and the way randomness is injected into the trees. Specifying a splitting leaf system involves

the collection of candidate split shapes and methods to determine the consistency of each candidate. Using axis aligned splits are common choices here, where a linear combination of characteristics eats thresholds to make decisions. In either case, the threshold value can be selected randomly or by optimizing the leaf's data function. A set of candidate splits are created in order to divide a leaf and a criterion is evaluated to choose between them.

A basic strategy, as in the models studied in Biau et al, is to select between the candidates uniformly at random (2008). The option of the candidate split, which optimizes a purity function over the leaves that would be generated, is a more popular approach. A typical decision here is to optimize the acquisition of knowledge (Hastie et al., 2013). In leaf, the most common choice for predictions is to use the average reaction over the training points that fall through that leaf. The use of several different leaf predictors for regression and other tasks is discussed by Criminisi et al. (2011), but these generalizations are outside the reach of this document. Here, we only consider simple average predictors. Injecting randomness into the construction of the tree can occur in several ways. It is possible to randomize the choice of which dimensions to be used as split candidates on each leaf, as well as the choice of coefficients for random combinations of features. In either case, it is possible to select thresholds either randomly or by optimization over some or all of the leaf data. Building each tree using a bootstrapped sub-sampled data set is another common method for implementing randomness. Each tree in the forest is trained in this way on slightly different data, which introduces variations between the trees.

In order to minimize the variance, random forests are a way of averaging several deep decision trees, trained on different sections of the same training set. This comes at the cost of a slight increase in bias and some lack of ability to perceive, but usually improves the output in the final model considerably (Kishan Maladkar, 2018). And according to Jianguo Chen, Kenli Li,Zhuo Tang,Kashif

Bilal, Shui Yu,Chuliang Weng and Keqin Li,2018, a suitable data mining algorithm for big data is the Random Forest (RF) algorithm. It is an ensemble learning algorithm that constructs the model using the feature sub-space. In addition, all decision trees can be trained simultaneously, so it is ideal for parallelization as well.

d. Support Vector Machine

Support Vector Machine is a supervised machine learning algorithm that can be used to classify or regress problems. The goal of the support vector machine algorithm is to find a hyperplane that classifies the data points in an N-dimensional space separately.

Based on the article written by T V Rampisela and Z Rustam, 2018, the method of Support Vector Machines is a binary method of classification aimed at generating a model with strong generalization capability with an optimal global solution. SVM generates a hyperplane that divides two target values to optimize the margin, the closest distance between the data and the hyperplane. Statistical learning theory was used to derive the SVM algorithm. The algorithm is based on

the structural risk minimization principle and can compact the array of raw data to
a support vector set (typically 3-5 percent of the raw data) and learn how to obtain a classification decision function. The basic principle is to build a hyperplane as the decision surface such that the maximum interval between the positive and the negative mode is (Jianfang Cao, Min Wang, Yanfei Li, Qi Zhang, 2019).

Support Vector Machine is one of the supervised algorithms for machine learning with outstanding effectiveness. SVM is a linear classifier based on the principle of maximization of margins and optimally uses the hyperplane to classify data into two classes of data in higher dimensional space. The margin is the distance between the hyperplane and the closest data from any class. The support

vector is called the closest data (Styawati Styawati and Khabib Mustofa, 2019). And among many predefined tasks, this task classifier will take a natural language text input and classify the input text into an implied task group, according to Hyungsik Shin and Jeongyeup Paek, 2018. Therefore, it can understand humans' natural language order and execute the intended task on behalf of the user accordingly. And in the journal written by Nick Gaunther and Matthias Schonlan, 2016, support vector machines are statistical and machine-learning approaches with the primary aim of prediction. Analogous to Gaussian, logistic, and multinomial regression, binary, and categorical outcomes can be continuously applied to them.

e. K-Nearest Neighbor

A simple algorithm that stores all available cases and classifies new cases based on a similarity measure is K nearest neighbors. As a non-parametric method, KNN was already used in statistical estimation and pattern recognition in the early 1970s. KNN is an approach to data classification that estimates how likely a data point is to be a member of one category or the other based on what group the data points closest to it are in. The k Nearest Neighbor (kNN) method is a popular classification method in data mining and statistics because of its simple implementation and substantial classification performance. However, it is impractical for traditional kNN processes to assign a fixed k value to all test samples (Shichao Zhang, Xuelong Li, Ming Zhong, Xionfeng Zhu, and Ruili Wang, 2018). And according to Zhongheng Zhang, 2016, by assigning them to the most comparable labeled examples class, the classifier kNN classifies unlabeled findings. Characteristics of findings are gathered for both preparation and assessment datasets. And the KNN algorithm is more adaptive to the input data of the local part which the classification problem makes it more unique (Harikumar Rajagru and Sannasi Chakravarthy, 2019).

Handayani, 2019, claimed that K-NN is a learning-based algorithm where the dataset training is stored. Therefore, the classification for the new record that is not graded is obtained by comparing the record that is most comparable to the training set. It is one of the most widely used classification algorithms in situations where there is little or no prior knowledge of data distribution. The core principle of KNN is to use a community of resilient neighbors in the training outcomes (S. Pandey, V. Sharma, and G. Agrawal, 2019).

2.2 Related Studies

In the study by Mohammed M. Ezz, entitled, Advisory Framework for University Student Enrollment Based on Variety of Machine Learning, Mohammed M. Ezz said that a predictive model for students would help him choose the best suitable faculty based on his grades for various high school subjects. For the student to make mature choices, the proposed model serves as an advisory and recommendation framework. The model can effectively assist faculty management in determining each student's main performance characteristics, and can therefore filter applicants based on smart predictive criteria.

Through machine learning, course and subject advising can be done automatically. According to Walid Mohamed Aly, et. al. in the study, Automated Student Advisory Using Machine Learning in 2013, educational data mining is a specific data mining field applied to data originating from educational environments. It relies on different approaches to discover hidden knowledge from the available data. Among these approaches are machine learning techniques which

are used to build a system that acquires hidden knowledge from previous data. To solve various regression, classification, clustering and optimization issues, machine learning can be applied. The research uses grouping and clustering for the Student Advisory System. The method can be used to direct university students from the first year to the most effective educational course. The classification stage

will predict the department that a student is most likely to select, and the clustering stage will suggest a department to the student by showing each department its expected success rate. The goal of this recommendation is to reduce the high rate of academic failure for students in the first year.

According to the study entitled, Automated Student Advisory using Machine Learning, by Walid Mohamed Aly, et. al., educational data mining field applied to data originating from educational environments relies on different approaches to discover hidden knowledge from available data. Among these approaches are machine learning techniques which are used to build a system that acquires hidden knowledge from previous data. Machine learning can be applied to solve different regression, classification, clustering and optimization problems. The Student Advisory Framework utilizes classification and clustering. This system can be used to guide the first year university students to a more suitable educational track. The classification phase will predict the department which is most likely to be chosen by a student and the clustering phase recommends a department to students by showing the expected rate of success for each department. This recommendation aims to decrease the high rate of academic failure for first year students.

Student advising is one of the many vital service roles that faculty have in academia, based on the study by Arun N. Nambiar and Anish J. Dutta entitled, Expert Framework for Student Advising using Jess. However, students also have very similar concerns about which classes to take, the course series, and deadlines. If programs have a striated system of student rankings, students are further confused as they may not be able to take such classes because of their current program status. Faculty also tend to be pressed for time with research, teaching and other committee responsibilities. This results in increased frustration for both students and faculty alike. Expert systems are software applications that respond to user queries by analyzing data captured in a knowledge system. JESS is a Java-

based rule engine and scripting environment that allows the development of such an expert system. In this work, develop an expert system using JESS that allows students to study to seek quick responses to their queries regarding their plan of study and progress in the program. This expert system separates the rules from the execution thus enabling users to customize or extend the system by changing or updating the XML file that stores the rules.

Building a Smart Academic Advising System Using Association Rule Mining by Raed Shatnawi et. al., in a study entitled Student advice is considered a paramount activity for both advisors and students to improve the academic performance of students in an academic setting. Advising is a time-consuming activity in universities with large numbers of students that can take considerable effort for advisors and university administration to guide students to successfully and efficiently complete their registration. Current systems are traditional and depend greatly on the advisor's effort to find the best selection of courses to enhance the performance of the students. A smart system is required that can advise a large number of students each semester. A smart system that uses association rule mining to assist both students and advisors in choosing and prioritizing courses has been proposed in this paper. The system has helped students enhance their performance by proposing courses that meet their current needs and improve their academic performance at the same time. To find associations between courses that have been registered by students in many previous semesters, the system uses association rule mining. A list of association rules that guide a particular student to select courses registered by similar students is successfully generated by the system.

According to the study entitled, A Proposed Academic Advisor Model Based on Data Mining Classification Techniques, university and higher institute admission are an intricate decision process and it is an important responsibility of the students to select the correct study track. The increase of the student's major

dropout rate in the higher education system is one of the important problems in most institutions. One approach to solve such problems and succeed in academic life is to help the students in selecting a suitable major and assign them to the right track. The objective of our research is to build an academic advisor model for students for their higher education which utilizes classification data mining for recommending the suitable academic major. The method applied in the research is data mining classification techniques through decision tree methods for advising students to select suitable majors and help assign them to the right track.

The proposed model classifies students and matches them to the proper study tracks according to their features. The three decision tree classification algorithms, namely J48, random tree and reduced error pruning (REP) tree were first applied to real data in a managerial higher institute in Giza Egypt and results were compared between them. Finally, the results showed that the J48 algorithm gives 16 rules and the rules that gave low CGPA were eliminated. The J48 algorithm gave the highest accuracy. The study then suggests using the generated J48 decision tree in the proposed student advising model to enhance student's academic performance and decrease dropout.

Based on the study, An Intelligent Student Advising System Using Collaborative Filtering by Kathiravelu Ganeshan and Xiaosong Li, the web-based intelligent student advising system using collaborative filtering, a technique commonly used in recommendation systems assuming that users with similar characteristics and behaviors will have similar preferences. With this advising system, students are sorted into groups and given advice based on their similarities to the groups. If a student is determined to be similar to a group of students, a course preferred by that case will be recommended to the students. K-means algorithm was used to determine the similarity of the students. This is an extremely efficient and simple algorithm for clustering analysis and widely used in data mining.

The study of Olawande Daramola et. al. entitled, Implementation of an Intelligent Course Advisory Expert System stated that academic advising of students is an expert task that requires a lot of time, and intellectual investments from the human agent saddled with such responsibility. In addition, good quality academic advising is subject to availability of experienced and committed personnel to undertake the task.

However, there are instances when there is paucity of capable human advisers, or where qualified persons are not readily available because of other pressing commitments, which will make system-based decision support desirable, and useful. In this work we present the design, implementation of an intelligent Course Advisory Expert System that uses a combination of rule based reasoning (RBR),and case based reasoning (CBR) to recommend courses that a student should register in a specific semester by making recommendations based on the student's academic history. The evaluation CASE yielded satisfactory performance in terms of credibility of its recommendations, and usability.

Samay Abu Naser et. al. stated in their study entitled, Predicting Student Performance Using Artificial Neural Network: in the Faculty of Engineering and Information Technology, that this model is for predicting the performance of a sophomore student enrolled in engineering majors in the Faculty of Engineering and Information Technology in Al-Azhar University of Gaza. A number of factors that may possibly influence the performance of a student were outlined. Such factors as high school score, score of subject such as Math I, Math II. Electrical Circuit I, and Electronics I took during the student freshman year, number of credits

passed, student cumulative grade point average of freshman year, types of high school attended and gender, among others, were then used as input variables for the Artificial Neural Network model. A model based on the Multilayer Perceptron Topology was developed and trained using data spanning five generations of graduates from the Engineering Department of the Al-Azhar University, Gaza.

Test data evaluation shows that the ANN model is able to correctly predict the performance of more than 80% of prospective students.

The Random Forest method builds on decision trees. A large number of decision trees is created by sampling individuals and variables in the training dataset. A key difference from the decision tree is that each node is split by the best of a random subset of variables, rather than the best of all the variables (Liaw and Wiener, 2002). Each individual is classified by each tree and the most common outcome is used as the final classification method including discriminant analysis, which is used extensively in forensic ancestry estimation (Liaw and Wiener, 2002). Another advantage of random forest classification is that it is robust to overfitting (Breiman, 2001; Liaw and Wiener, 2002). In this analysis, 500 decision trees were produced by sampling the training dataset with replacement (bootstrapping). Missing values were replaced by the modal value for the trait during the model-building phase. Once a model was built the validation datasets were then classified using the random forest. For model validation, individuals with missing data were omitted if they could not be classified by all the decision trees in the random forest. The random forest models were built using the random forest package (Liaw and Wiener, 2002).

A useful result obtained from random forest modelling was relative variable importance. In this analysis the Mean Decrease Gini based on the Gini impurity index was used. Each split at a node (non-metric trait variable) should increase the homogeneity (purity) of the two descendant nodes resulting in a reduction of the Gini impurity criterion. The average decrease in the Gini for each variable overall the trees gives an accurate picture of the relative importance of the variables in the model. However, one should keep in mind that true variable importance can be obscured by complex variable interactions (Breiman, 201l).

Based on the study entitled “The impact of predictive analytics based advising on the selection and change of major among first year, first-term students in engineering" by Sylvester Charles Upah,2016, to evaluate which variables were important in predicting results and to notify which variables were important, logistic regression models were constructed. Listed for use in propensity score tests to assess the effect on student outcomes of the procedure. Logistic regression is a methodology that researchers in the field of educational data mining have commonly used. Observing the success of students in previous semesters typically influences their potential performance (Li Zhang, 2018).

In the study entitled Data Mining for Student Advising the authors Hosam Alhakami, Tahani Alsubait, Abdullah Aljarallah, 2020, stated the neural network uses the biological nervous system's gradient descent method, which has many interrelated processing components. These components are known as neurons. The operability of the learned network is improved using the rules derived from the qualified neural network. The Neural Network is very reliable and is used to direct students at secondary level for subject/stream selection. During the creation of the model research would increase the output rate and decrease the dropout rate. The advantage of considering factors related to the interest of the student is to make this support structure well suited to effectively provide a solution from the point of view of the student keeping equal weight with other aspects discussed in previous sections (Kapil Sethi, Varun Jaiswal, Mohd Dilshad Ansari, 2018).

A robust and precise technique for pattern classification and information mining, the multi-class support vector machine (SVM) has attracted little attention from research in the educational field. This technique has been recorded to have a sound and highly reliable theoretical basis. The application of multi-class SVM technology to educational data, unlike other algorithms, is yet to be thoroughly explored, this statement was made by Mojisola G. Asogbon, Oluwarotimi W. Samuel, Mumini O. Omisore, and Bolanle A. Ojokoh in their study entitled "A Multi-class Support Vector Machine Approach for Students

Academic Performance Prediction", 2016. In the study "Predict Student Performance by

Utilizing Data Mining Technique and Support Vector Machine", writers Ankita Kadambande, Snehal Thakur, Akshata Mohol, and Prof A.M.Ingole, 2017, stated that in each lesson, the Support Vector Machine (SVM) is used to predict student grades. In order to further boost the results of the forecast, we apply a majority vote method to the predicted results obtained in consecutive lessons to conveniently keep track of the learning situation of each student.

According to a study entitled "Kernel-based Support Vector Machine for Rice Blast Disease Prediction in Northern Philippines" by Alvin R. Malicdem and Proceso L. Fernandez, 2014, SVM is a method based on the principle of predictive learning which has been used to solve problems of classification and regression. Intuitively, the problem of optimization constraints is formulated by collections of mathematical equations. A well-known learning model that is primarily used for classification and regression analysis is the Support Vector Machine (SVM). SVM calculates a maximum-margin line (or hyperplane) separator for the classification category of problems, which is the problem type for our analysis, which can properly classify the instances of each class (Jan Miles Co and Proceso Fernandez, 2017).

In the study entitled "Predicting River Pollution Using Random Forest Decision Tree with GIS Model: A Case Study of MMORS, Philippines" by Jayson M. Victoriano, Luisito L. Lacatan, and Albert A. Vinluan,2020, stated that, by constructing decision trees from a bootstrapped sample taken from a training set, Random Forest functions. B repeats this process several times where B is the optimal number of trees generated for the forest. A node is divided on the basis of the best among the random subset of features during the construction of a tree. Tree-based ensemble methods are known to be Random Forest and Gradient Boosting (GB), in which multiple tree sub-models are generated and combined to generate an enhanced final model. The distinction lies in the process of constructing several tree models between the two ensemble approaches. In generating a large number of independent bootstrapped trees at random from the

dataset, RF utilizes the bootstrap aggregation (bagging) ensemble method. Then the different models of the tree are aggregated or combined using the mean (Thaddeus M. Carvajal,Katherine M. Viacrusis, Lara Fides T. Hernandez, Howell T. Ho, Divina M. Amalin, and Kozo Watanabe.

According to Joenel G. Galupino and Jonathan R. Dungca, 2018, in their study titled “Quezon City Soil Profile Reference”, the predictions of k-NN are based on the intuitive assumption that objects close to distance are theoretically identical. When making predictions, it makes good sense to differentiate between the K nearest neighbors. And in the study entitled “Implementation of Template Matching, Fuzzy Logic and K Nearest Neighbor Classifier on Philippine Banknote Recognition System”, authors Rodel Emille T. Bae, Edwin R. Arboleda, Adonis Andilab, and Rhowel M. Dellosa, 2019, stated that KNN is a method that classifies objects in the feature space based on the closest training examples. It is the most fundamental method of example-based learning. In n-dimensional space, it assumes all instances are points. For determining the closeness of instances, a distance measure is required. By finding its nearest neighbors and choosing the most common class among the neighbors, KNN classifies an instance. One of the easiest of all machine learning algorithms is the k-Nearest Neighbor (k-NN) algorithm as it is easy to implement and only uses distance comparisons. However, when compared with other classification algorithms, the precision of k-NN is still relatively poor (Jasper Kyle Catapang, 2019).

According to Ms. Ghalia Musalam Salim Alfarsi, Dr. Khaled Abdalgader M. Omar, Ms. Maryam Juma Alsinani, authors of “A Rule-Based System for

Advising Undergraduate Students", 2017, the expert system (ES) is one of the most effective fields of artificial intelligence. It is a rule-based decision-making engine that allows non-expert users to improve their abilities. The ES is regarded as a framework that draws on the information obtained from experts. OCR (Optical Character Recognition) is a tool for translating written text into editable text. In

different applications, OCR is a very suitable and standard process. The correctness of OCR can be very dependent on the algorithms for pre-processing and segmentation of text (Cristopher C. Abalorio and Monalee dela Cerna, 2018). And in the study entitled "Implementation of an Intelligent Course Advisory Expert System Cased-Based Course Advisory Expert System" written by Olawande Daramola, Onyeka Emebo, Ibukun Afolabi, and Charles Ayo, 2014, they stated that CBR is a paradigm of pattern-based problem solving based on data learned from previous cases to solve new problems if it is possible to establish enough parallels between the current case (problem) and previous cases stored in the base case.

Chapter Three

THEORETICAL FRAMEWORK

Subject Advising

Traditional subject recommendation by the faculty members is a critical and complicated activity due to the process of thoroughly investigating individual cases of students, particularly irregular students of the Universidad de Manila. It inevitably requires human intervention as it requires to pass through decision making based on different factors. One of these factors includes the grades in previous subjects taken, significantly, the prerequisite subjects. The decision of a particular student to shift course or transfer to a different school is another factor that a dean or an equivalent position needs to take into consideration for subject advising. Also, the workload and the study hours of subjects for a single student is a significant factor for subject advising. This process generally takes a substantial amount of time that could only partake for a few concerned students.

Proposed Model and Process

COURSE CODE	COURSE TITLE	Credit Units	# of hours LEC	# of hours LAB	Prerequisite / Corequisite
FIRST YEAR, First Semester					
CSC111	INTRODUCTION TO COMPUTING	3	2	3	
CSC112	COMPUTER PROGRAMMING 1	3	2	3	
GECS01	UNDERSTANDING THE SELF	3	3		
GECS02	READINGS IN PHILIPPINE HISTORY	3	3		
GECS03	THE CONTEMPORARY WORLD	3	3		
GECS04	MATHEMATICS IN THE MODERN WORLD (MATH)	3	3		
PE1	MOVEMENT ENHANCEMENT	2	2		
NSTP1	NSTP1	3	3		
	TOTAL	**23**	**21**	**6**	
FIRST YEAR, Second Semester					
CSC121	COMPUTER PROGRAMMING 2	3	2	3	CSC112
MS101	DISCRETE MATHEMATICS	3	3		
GECS05	PURPOSIVE COMMUNICATION	3	3		
GECS06	ARTS APPRECIATION	3	3		
GECS07	SCIENCE, TECHNOLOGY AND SOCIETY	3	3		
GECS08	INTRO TO ETHICS	3	3		
PE2	FITNESS EXERCISES	2	2		
NSTP2	NSTP2	3	3		
	TOTAL	**23**	**22**	**3**	
FIRST YEAR, Summer					
CSC211	DATA STRUCTURES & ALGORITHMS	3	2	3	CSC121
ITE122	WEB DEVELOPMENT	3	2	3	
ITE121	INTRO TO HUMAN COMPUTER INTERACTION	3	2	3	
	TOTAL	**9**	**6**	**9**	
SECOND YEAR, First Semester					
CSC212	OBJECT ORIENTED PROGRAMMING	3	2	3	CSC211
ITE211	DATABASE MANAGEMENT SYSTEMS 1	3	2	3	
ITE212	OPERATING SYSTEMS	3	2	3	
ACT211	FUNDAMENTALS OF ACCOUNTING	3	3		
GEMS01	THE LIFE AND WORKS OF RIZAL	3	3		
GECS09	KONTEKSTWALISADONG KOMUNIKASYON SA FILIPINO	3	3		
PE3	PHYSICAL ACTIVITIES TOWARDS FITNESS 1	2	2		

Figure 3.1 Sample Program of Study

Each student is given a program of study in their first year of study. The program of study identifies the set of subjects they need to take for each year and each semester. It also shows the list of units, lecture and laboratory hours, and the prerequisites of each course.

Minimum Spanning Tree

This plan or program of study can be represented in a network diagram, specifically a directed graph. The nodes represent the subject and the edges represent the prerequisites or dependencies of each subject to one another. The diagram presented in figure 3.1 represents the directed graph of the BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY (BSIT) curriculum of the university.

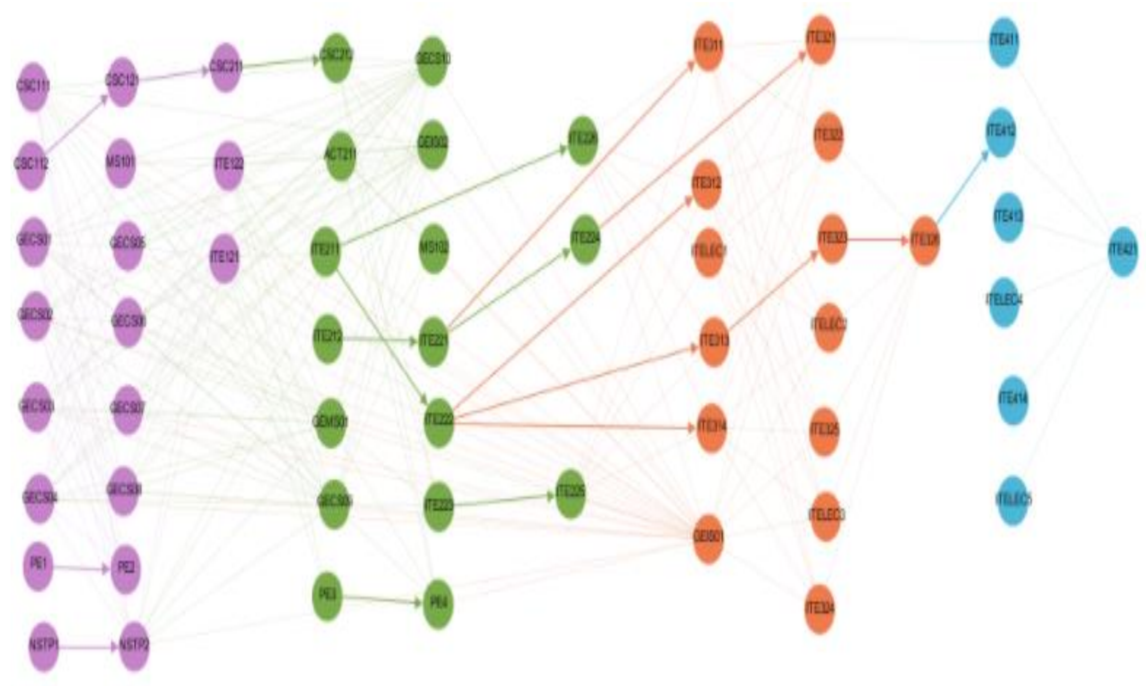


Figure 3.2 Directed Graph of Prerequisite Mapping for BSIT Program

Minimum Spanning Tree (MST) is utilized to represent the program of study of the students. MST is used for approximating the traveling salesman problem. This problem is similar to how the program of study can be represented. Since each subject on the map is required to be taken by the student, it is mandatory that all nodes should be visited. The ultimate goal is to identify the shortest path while visiting all courses. The cost of the spanning tree is the sum of the weights of all the edges in the tree. There can be many spanning trees. The minimum spanning tree is the spanning tree where the cost is minimum among all the spanning trees. There also can be many minimum spanning trees. There are numerous combinations of paths that can be identified in each program. The minimum spanning tree represents the combination of courses that could be easier for the student, giving them higher chances of finishing the program on-time. It further becomes more complex with the irregular students. Numerous constraints need to be considered. These include the maximum number of units that can be taken each semester, the lecture and laboratory hours, and the prerequisite courses. To identify the shortest path, each edge connecting each node should be labeled or assigned by a certain value - the weights.

Kruskal's Minimum Spanning Tree Algorithm

Kruskals' algorithm is a method in finding the minimum spanning tree of a given weighted graph. It takes graph as an input, finds the subset of the edges that forms a tree which includes every vertex, and has the minimum sum of weights among all the possible formed trees from the graph.

The Kruskal's Minimum Spanning Tree algorithm is shown below:

1. Sort all the edges in non-decreasing order of their weight.
2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include its edge. Else, discard it.
3. Repeat step#2 until there are (V-1) edges in the spanning tree.

Weights

To implement the concept of minimum spanning tree in recommending courses to be taken by the students, it is necessary to provide weights on each edge connecting the nodes or courses. Weights on each edge will be calculated using a machine learning algorithm as proposed in this study.

Table 3.1 Sample Subject Weight Assignment

COURSE CODE	PREREQUISITE	WEIGHT

ITE223		95.00
PE4		94.00
ITE224	ITE221	88.00
ITE225	ITE223	87.00
ITE226	ITE211	87.00
ITE311	ITE221	83.30
ITE312	ITE222	81.00
ITELEC1		78.70
ITE313	ITE222	76.40
ITE314	ITE222	74.10
GEIS01		71.80

Presented in table 3.1 is the sample representation of how weights were assigned on each course. This is fundamental to the identification of the shortest path, and further suggest courses which are optimally necessary to be taken by a specific student.

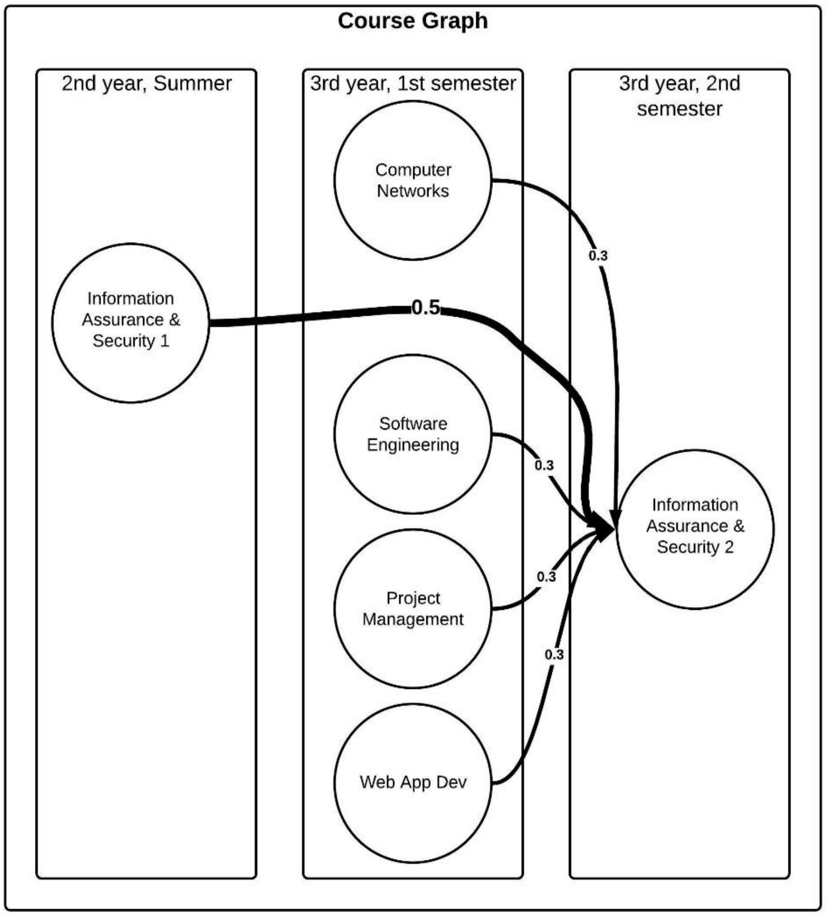


Figure 3.3 Course Graph Weighing

To further illustrate, figure 3.3 shows a sample of weighted notes across all available courses. These weights associated with each course were computed based on the algorithm presented in the machine learning model. Those with higher weights have higher scores in terms of recommendation. Based on the maximum number of allowed units to take for a specific semester, the courses with the highest scores will be recommended to the student in course advising.

Machine Learning Model Development

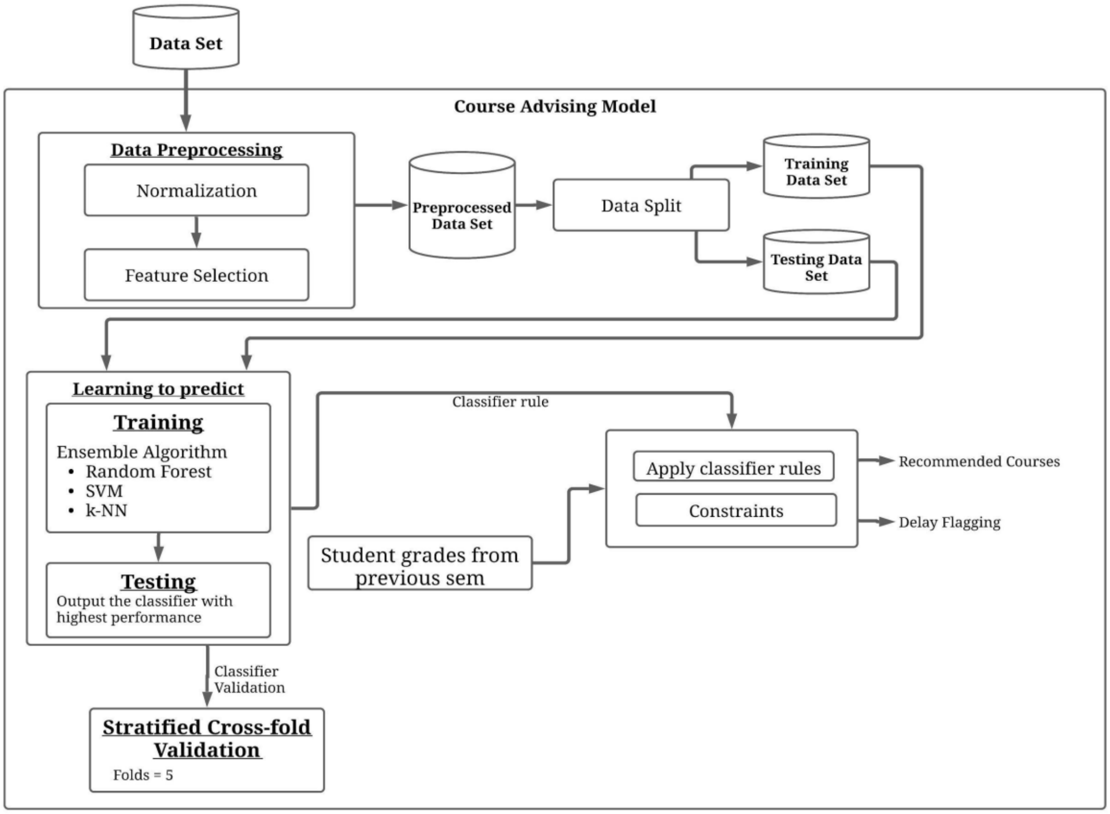


Figure 3.4 Machine Learning Model Phases

To deeply understand how the model was developed, Figure 3.4 shows the phases of the machine learning model for automated course advising. The data set loaded in the model has undergone normalization. Features from the normalized data were selected and were splitted as training and testing data sets. Training data sets were fitted to an ensemble the algorithm that combines three different algorithms: Random Forest, Support Vector Machine (SVM), k-Nearest Neighbor (k-NN). The algorithms were tested to output the classifier with the highest performance. Before the classifier was used, it has undergone evaluation using the Stratified Cross-fold Validation where the number of folds is equal to five. The classifier is embedded in the classification rules and constraints phase of the model as a filter for the input data, specifically, the grades of a student from the previous semester. The model produces the recommended course of the student and will identify the student's condition of his academic degree.

Data Sources

The student data is collected from Universidad de Manila during the period from 2015 to 2020 to form the data set. This data set includes 1, 237 records, each record has 15 attributes. Not all the attributes will be used in the data mining process, some of the attributes in the data set such as the Student ID, Student Name, Address, or Home Phone Number present personal information that does not expand any knowledge for the data set under processing.

Data Preprocessing and Cleanup

It is ultimately necessary to preprocess the raw data before the development of the actual model. The data preprocessing stage takes 60% of the model development process. It is important that individual data are carefully examined to remove possible anomalies and produce an accurate model. The data preprocessing stage includes feature selection and anomaly detection.

Feature Selection

Since not all fields from the raw data are necessary for the development of the model, the fields were trimmed down and irrelevant ones were deleted. The selected attributes are shown in Table 3.2.

Table 3.2 Selected Feature for ML Model

Attribute	Description	Data Type	Range
Course Code	Uniquely identifies a course	Discrete	N/A
Units	Total number of units	Continuous	1-10
Semester	Semester where the course is associated with	Discrete	1-3
Year Level	Year Level where the course is associated with	Discrete	1-5
Grade	Grade earned by a student on a specific course	Continuous	0-100
Lec Hours	Lecture hours for a specific course	Discrete	1-5
Lab Hours	Laboratory hours for a specific course	Discrete	1-5

Anomaly Detection

The data produced for this research was produced by the records department of the university. As much, there is no complete assurance that the given data were free from anomalies or inconsistencies. As part of the data preprocessing stage, the researcher manually examined the individual files to identify anomalies such as records with incomplete values, null values for fields that are not nullable, data consistency on course codes, and range inclusion on submitted grades.

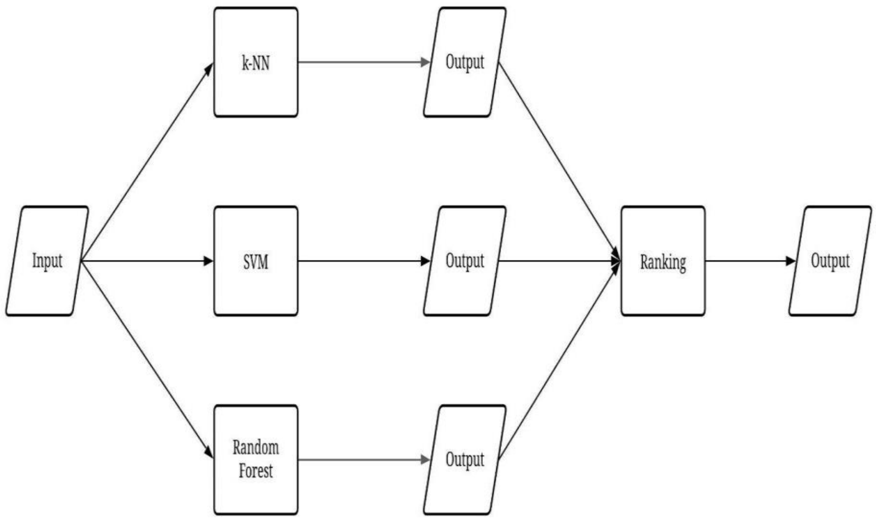
Model Development

Ensemble Algorithm

The Ensemble approach will be used to improve the accuracy of subject advising. Ensemble methods are techniques that create and then combine several models to achieve higher accuracy. Ensemble methods typically offer more

accurate results than a single model can provide. Listed below are the 3 models that will be used.

- Random Forest
- Support Vector Machine
- K-Nearest



- Neighbor

Figure 3.5 Ensemble Algorithm Application

Random Forest

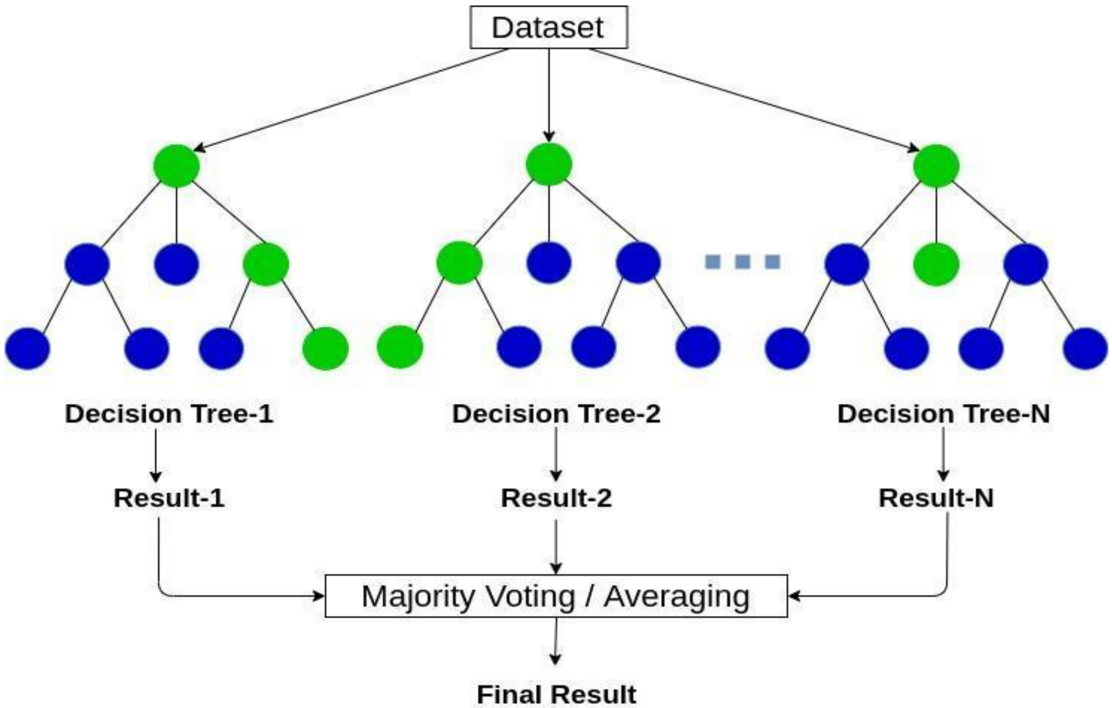


Figure 3.6 Random Forest Classifier Process

Image from Sharma, A., 2020. *Decision Tree Vs. Random Forest - Which Algorithm Should You Use?*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/> [Accessed 7 November 2020].

Random forest, an ensemble method, is a supervised learning algorithm that constructs and merges multiple decision trees to get a more precise and accurate prediction. Random Forest searches for the best feature among a random subset of features. It utilized the bagging method that combines learning models to increase the overall result.

Support Vector Machine

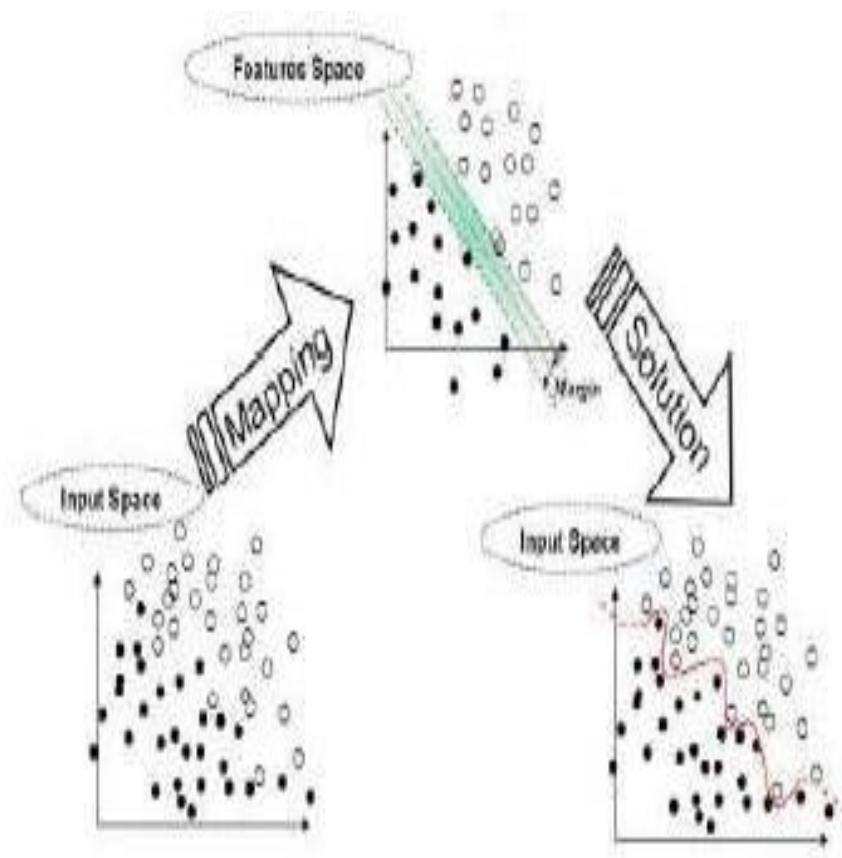


Figure 3.7 Support Vector Machine Process

Image from Denmag., 2015. *Collection of SVM Libraries By Language*. [online] Data Science Central. Available at: <https://www.datasciencecentral.com/profiles/blogs/collection-of-svm-libraries-by-language> [Accessed 7 November 2020].

SVM performs classification by finding the hyper-plane that differentiates the two classes very well. It transforms complex data, then works out how the information is separated based on the labels or outputs that have been defined

K-Nearest Neighbor

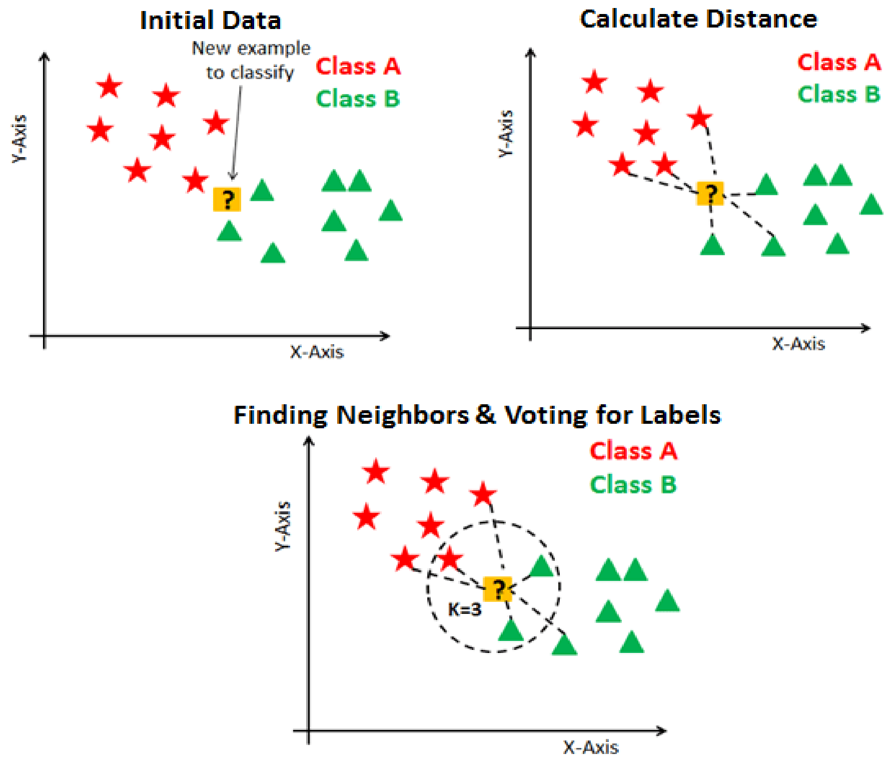


Figure 3.8 k-Nearest Neighbor Algorithm

Image from Chauhan, N.S., 2019. *Classifying Heart Disease Using K-Nearest Neighbor*. [online] KDnuggets. Available at: <https://www.kdnuggets.com/2019/07/classifying-heart-disease-using-k-nearest-neighbors.html> [Accessed 7 November 2020].

KNN performs by finding the difference between the query and all the data examples, identifying the defined number of examples (K) nearest to the query, and determining the most frequent in the case of classification or average in the case of regression marks.

After running through the three different algorithms presented, the mean scores will serve as a final identifier of whether a course will be included in the list of recommended courses for advising or not. This process encompasses the ensemble approach.

Table 3.3 Ensemble Ranking and Final Recommendation Selection

Course	KNN	SVM	Random Forest	Mean Score	Rank
ITE312	94.00	95.02	96.04	95.02	1
ITE223	93.00	94.29	95.59	94.29	2
ITE311	93.00	93.99	94.98	93.99	3
ITELEC1	92.00	93.02	94.04	93.02	4
ITE314	91.00	92.01	92.34	91.78	5
ITE313	90.00	90.99	91.98	90.99	6
ITE226	90.00	90.97	91.94	90.97	7
ITE224	85.00	86.04	87.07	86.04	8
ITE225	82.00	82.91	83.82	82.91	9
PE4	72.00	72.85	73.69	72.85	10

To illustrate, table 3.3 shows the scores gathered of the available courses not yet taken by the student. Each algorithm scored the courses and their mean scores served as the final basis. The first five highest-ranking courses were selected as a maximum unit is a constraint that needs to be considered.

Stratified Cross-fold Validation

To ensure the effectiveness of the developed classification model, Stratified Cross-fold Validation was used. This approach helps evaluate the performance of the classifier and avoid the overfitting of the model.

Table 3.4

Test Prediction Results

Prediction	
Correct	**Incorrect**
148	28

The Stratified Cross-fold Validation divides and distributes the dataset into different stratas. For every n stratas of the dataset, there are also n folds which correspond to the n times alternate training of the n-1 strata and testing it to the holdout n^{th} test strata. The sample process as shown in figure 3.9.

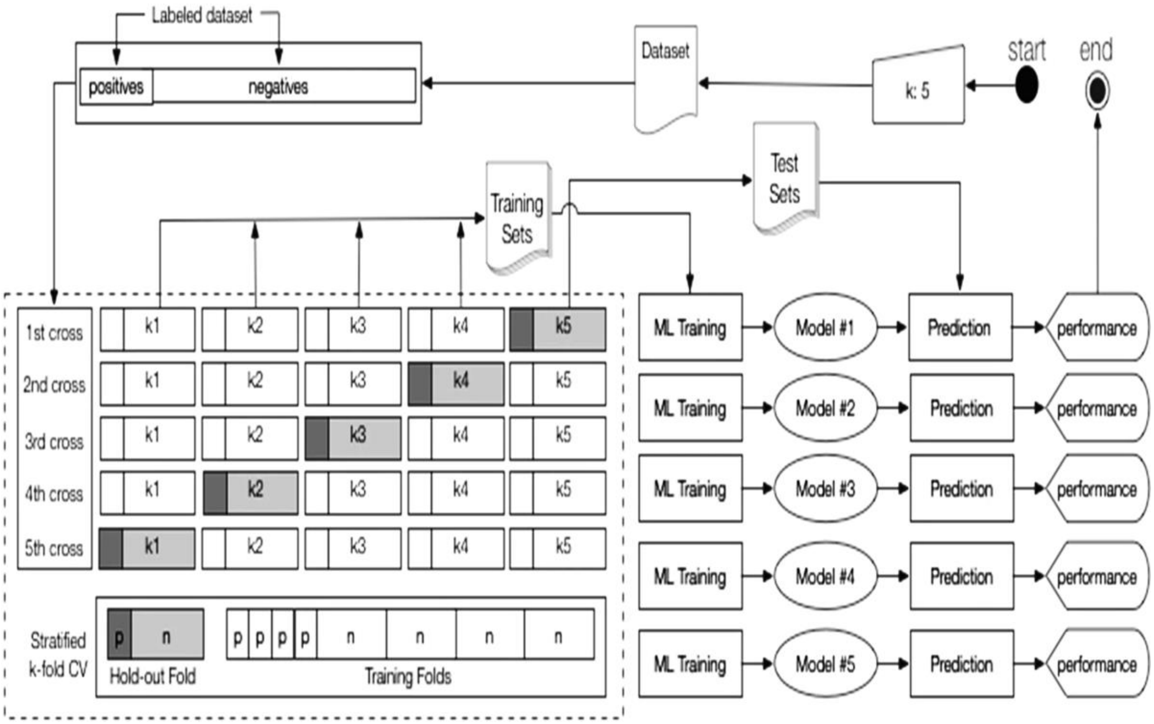


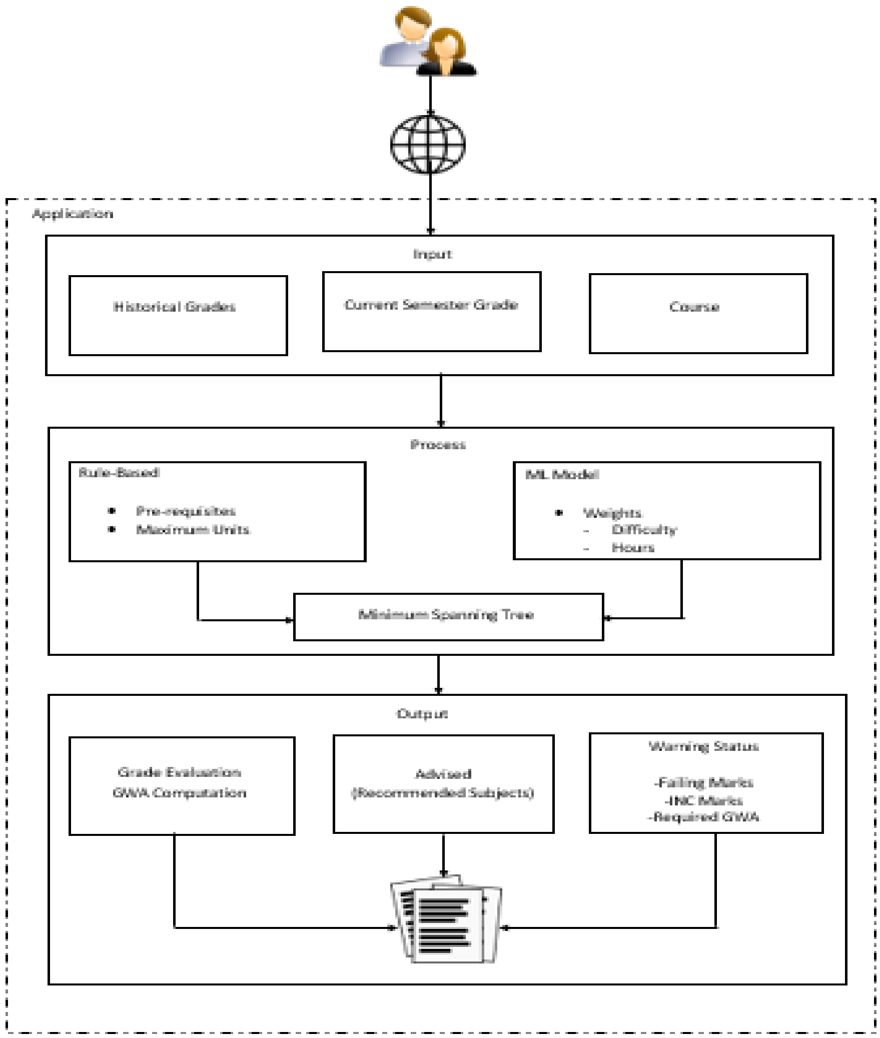
Figure 3.9 Stratified cross-fold validation of 5-folds

Image from Hasanin, T., Khoshgoftaar, T.M., Leevy, J.L. *et al.* Investigating class rarity in big data. *J Big Data* 7, 23 (2020).

Conceptual Framework

This section illustrates the totality of the final product of this research. It defines the relevant variables used in this study and maps out how they might relate to each other. This includes all the underlying concepts and their associated mappings based on the system's use.

Figure 3.10 Conceptual Framework



The model presented in figure 3.10 shows the flow of data to illustrate how the recommendations are generated.

The users of the proposed application are the faculty members and can be potentially rolled out to students for independent/self-service subject advising. In this way, the utilization of the faculty members involved in the process can be freed up. The proposed application is designed to work on desktop and laptop computers, mobile phones, and tablets, independent from any web browsers.

The input itemizes the list of parameters needed by the application and will fundamentally serve as a basis to weigh in recommendations. The process includes the combination of rule-based constraints and the ML model scores. With the algorithm therein, the application will be able to generate the list of recommended courses.

The proposed application is projected to perform the following processes: subject advising, status flagging to send an alert on the possibility of delay in finishing the program on time, grade evaluation, and academic status determination of the students. The proposed application can generate the list of subjects to be enrolled by a student. ML-based Weighted Minimum Spanning Tree will be utilized for subject advising.

Chapter Four

METHODOLOGY

This chapter is a description of the methods chosen to achieve the objectives of the study. It describes the respondents and the research instruments that were utilized in the study. It goes on to describe the techniques of data collection that aided in the completion of the research project.

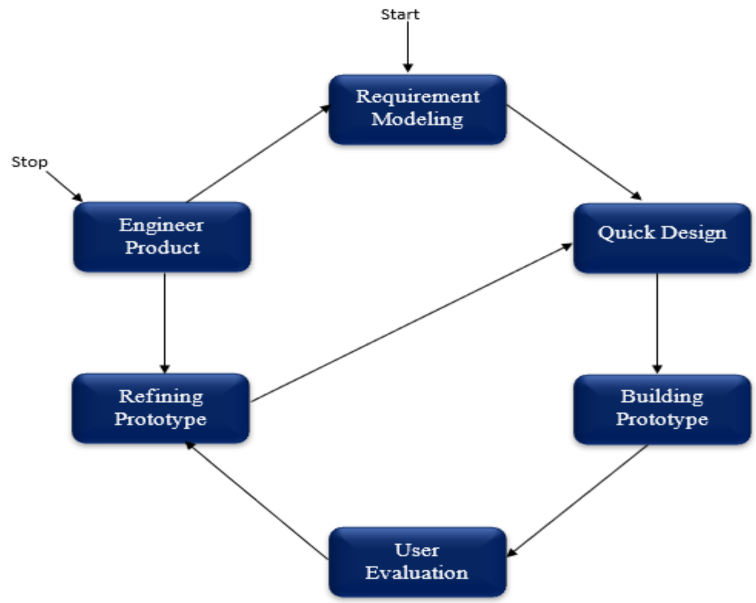


Figure 4.1 Prototype Model

Figure 4.1 shows the Prototype Model used by the researcher as a guide in developing the proposed study entitled An Intelligent Student Advising System: A Minimum Spanning Tree Approach which is an example of the System Development Life Cycle (SDLC) model. Prototype model is used because this is the model that best suits the client's needs and requirements. Using this model, it will enhance the usability, design quality of the proposed application and it will also make the development process more cost-efficient since the development cycle becomes shorter.

The phases of the prototype model involves the following steps:

4.1 Requirements Modeling

The Prototype Model begins with the outline requirements. In this stage, a preliminary investigation was conducted to identify the purpose and the use of the application together with the nature and scope of the study. It is also in this stage that the researcher asked permission from the school authorities to conduct the study and all pertinent data and information were analyzed.

Fact-finding was used through interviews to develop the logical model of the application. Information was gathered through these interviews and document analysis. Through these methods, the researcher was able to gather needed requirements for the development of the proposed application, particularly on how the application will work.

The study is conducted at Universidad de Manila which is located at Mehan Gardens, Manila. It is a public university that is funded by the local government unit of Manila, together with Pamantasan ng Lungsod ng Maynila. It was established in 1995, through the initiative of Mayor Alfredo S. Lim. The university offers courses from bachelor to postgraduate degree programs. It also offers Senior High School. There are eight colleges which include the College of Arts and Sciences, College of Business Accountancy and Entrepreneurship, College of Criminal Justice, College of Education and Human Kinetics, College of Engineering and Technology, College of Public Administration, College of Law, and College of Health and Sciences.

At present, the university is managed by Ms. Maria Lourdes N. Tiquia, as the president. There are 8, 163 students who are presently enrolled and 120 faculty members including the deans. The researcher interviewed selected faculty members of the different colleges of the university. The faculty members are considered as the main clientele of the said study. Since they are the ones responsible for advising students on what subject to

take during enrollment specifically the irregular students. Part of the assistance rendered by the faculty members during enrollment is grade evaluation and subject advising.

The current subject advising of the students of the Universidad de Manila is found to be done manually which is one of the tedious tasks of a faculty member during enrollment. Though the university has an existing online-based enrollment, it is not capable of advising irregular students on what subjects they are to enroll in for the next semester. It is said that subject advising is a crucial responsibility of the faculty members because wrong advice may lead to non-graduation of the student. Subject advising is only applicable to students who have failed a subject that is a prerequisite of another subject. When a student fails the latter, he shall not be allowed to enroll in the subject that requires it. He can only retake the failed subject if it is offered once again.

It is for these reasons that the researcher opted to design and develop an application that is fast, accurate, reliable, and easy to use. This application is projected to help ease the burden of the faculty members who are handling subject advising during the enrollment period by providing accurate information.

The data used in the study is a historical academic data of students from Universidad de Manila. Each of the historical data of the students includes courses taken, information about a course and the student's remarks for each course. Only courses with proper remarks, whether passed or failed, were acquired. After collecting the data needed, information about each course was cleaned by correcting its corresponding course code, units, lecture hours, lab hours, and course prerequisite/s. Also, proper spelling and value range for the remarks were thoroughly checked. Another information was added to each course, it was the expected year and semester a course should be taken, as a numerical value based on a program's curriculum checklist.

In implementing an automated subject advising, a model is used which is the Kruskal's Minimun Spanning Tree. The algorithm starts with weighted edges. It take all

the edges and sort them from lowest to highest based on the weight of the edge. Pick the smallest weighted edge from the sorted edges, if there is more than one, choose from any of it. Check if it formed a cycle with spanning tree, if the cycle is not formed then include the edge into the spanning tree, else, discard. Repeat the process of selecting the smallest weighted edge that is not part of the spanning tree until there are v-1 edges. Figure 4.2 shows the pseudocode of this algorithm.

```
KRUSKAL(G):
A = Ø
For each vertex v ∈ G.V:
    MAKE-SET(v)
For each edge (u, v) ∈ G.E ordered by increasing order by weight(u, v):
    if FIND-SET(u) ≠ FIND-SET(v):
    A = A ∪ {(u, v)}
    UNION(u, v)
return A
```

Figure 4.2 Kruskal's Algorithm

The procedure of Kruskal's algorithm could be described in the following steps:

1. Sort all the edges in non-decreasing order of their weight. The syntax below shows how the edges are sorted.

   self.graph = sorted(self.graph,
   key=lambda item: item[2])
   parent = []
   rank = []

   To create V subsets with single elements, the code is shown below:

   for node in range(self.V):
   parent.append(node)
   rank.append(0)

To determine the number of edges to be taken, the syntax below is applied:

while e < self.V - 1:

To illustrate how Kruskal's algorithm works on the Bachelor of Science in Information Technology program. First, consider graph G and its edge E with weights in Figure 4.3 with CSC112, CSC111, GECS04, MS101, CSC121, and GECS07 as vertices. Second, sort the weights of the edges in ascending order.

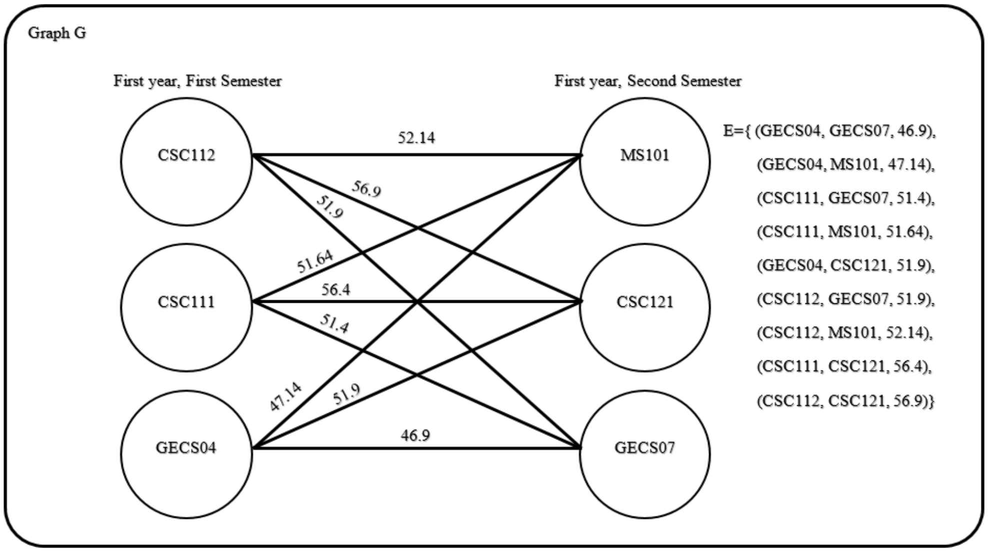


Figure 4.3 Graph G

2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include its edge. Else, discard it. This step will be repeated until all edges from the result of V-1 are plotted. The code is shown below:

```
u, v, w = self.graph[i]
# print(self.graph[i])
i = i + 1
x = self.find(parent, u)
y = self.find(parent, v)
# print(y)
```

This code is being applied when the plotting of the weights is forming a cycle with this code it will discard that specific edge that causes the formation of a cycle.

```
if x != y:
 e = e + 1
 result.append([u, v, w]
 self.union(parent, rank, x, y)
```

To illustrate, take the lowest weighted edge as the first branch of the spanning tree as shown in Figure 4.4.

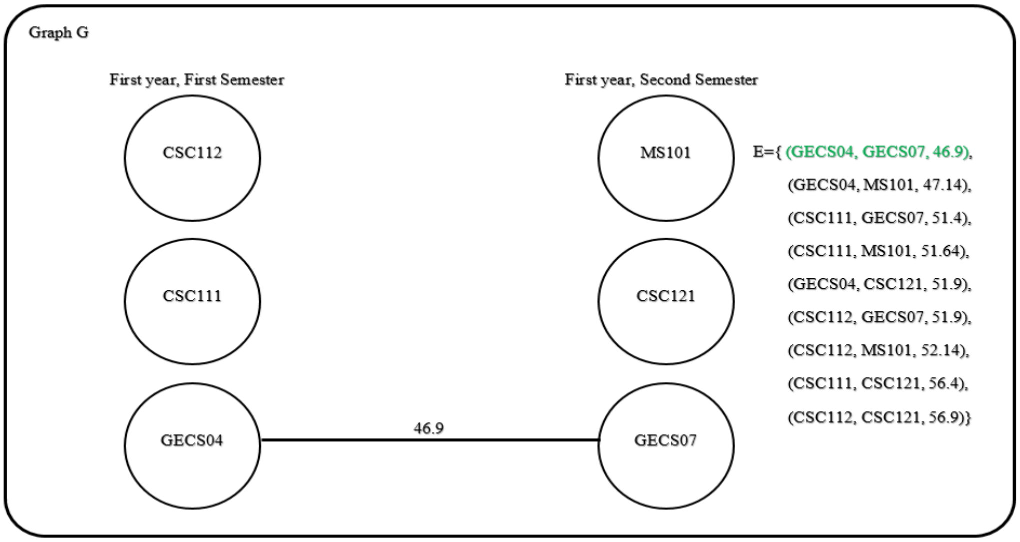


Figure 4.4 First Branch of the MST

After that, include the second lowest weighted edge into the spanning tree that is shown in Figure 4.5.

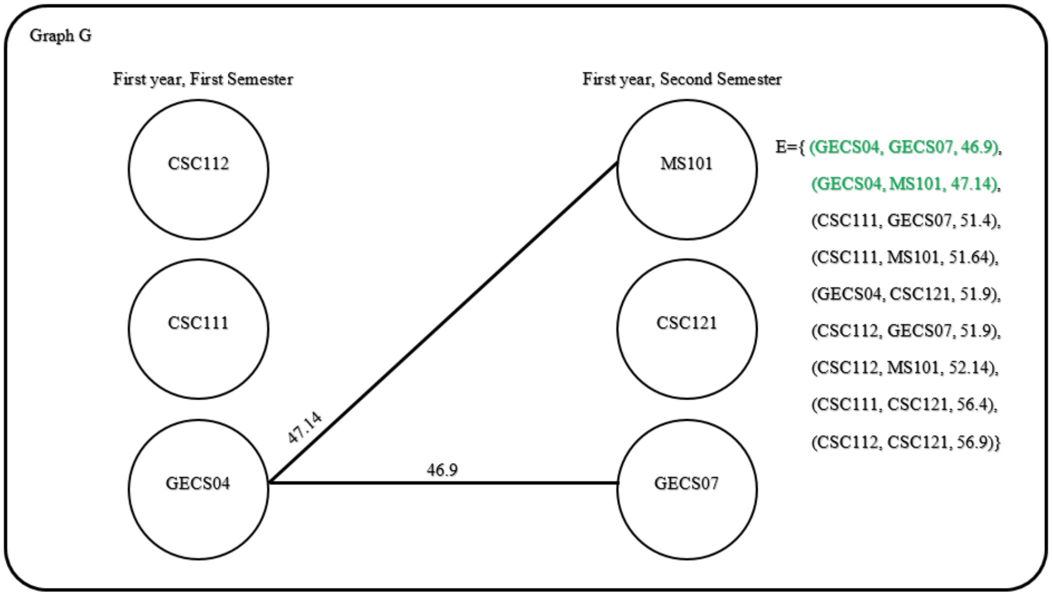


Figure 4.5 Second Branch of the MST

Include the next edge with lowest weight that is not yet in the minimum spanning tree just as in Figure 4.6.

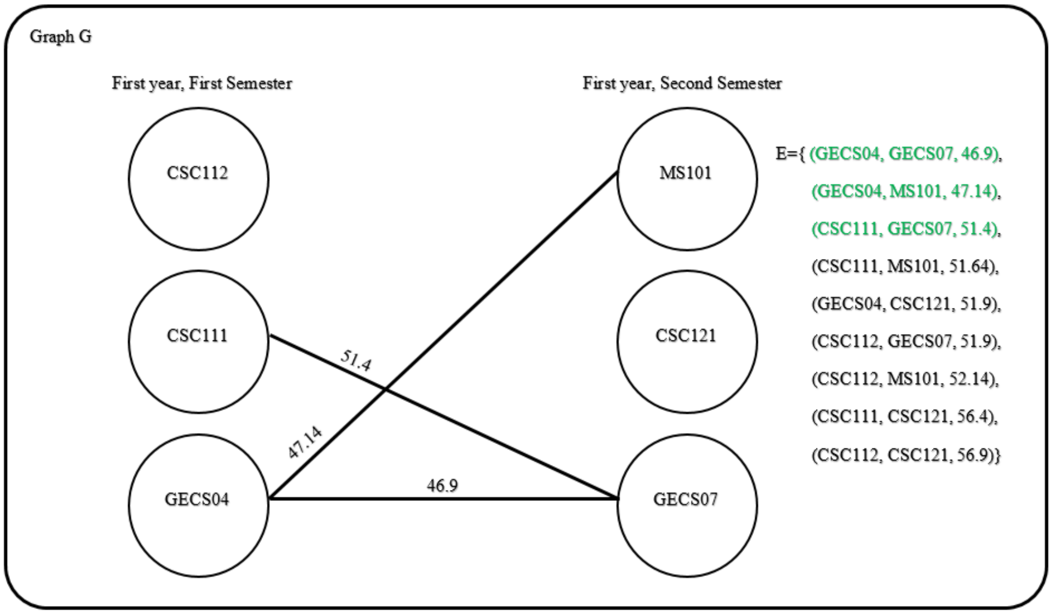


Figure 4.6 Third Branch of the MST

Figure 4.7 shows the fourth weighted edge in the sorted edges. It cannot be part of the minimum spanning tree because it creates a cycle of vertices CSC111, GECS04, MS101, and GECS07.

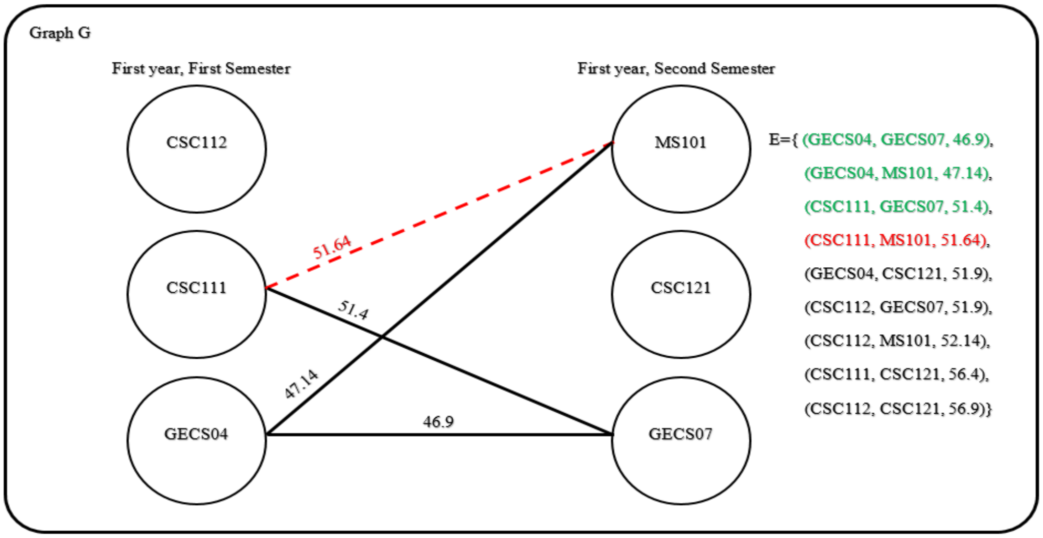


Figure 4.7 Invalid Edge

In the next branch of the tree it can be either of the two edges with equal weights. In Figure 4.8, edge GECS04-CSC121 is selected to be included in the minimum spanning tree.

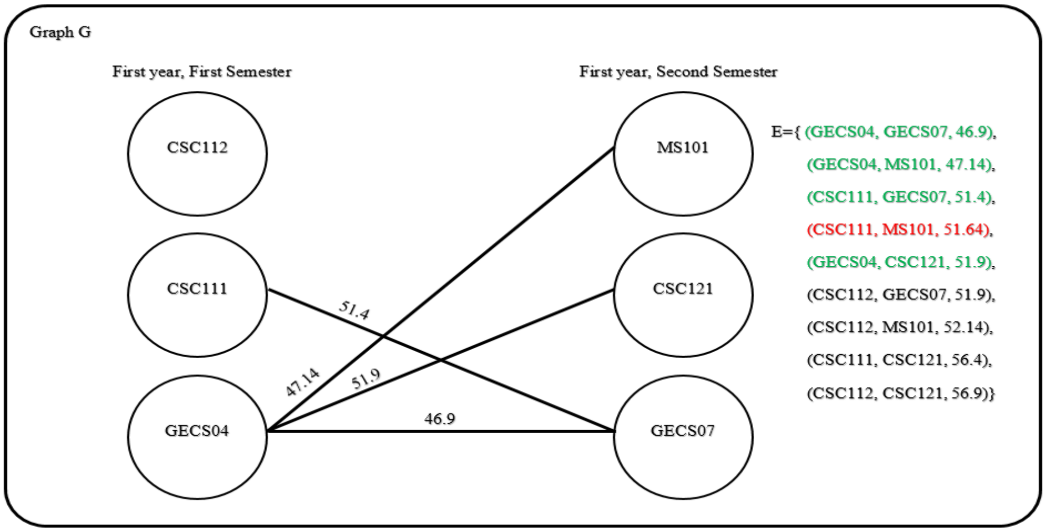


Figure 4.8 Fourth Branch of the MST

3. Repeat step#2 until there are (V-1) edges in the spanning tree. The code below will display the Minimum Spanning Tree formed from step1 to step 2. This will include the parent node, its destination and the computed total weight of the two nodes.

```
minimumCost = 0
nodes1 = []
nodes2 = []
weights_node = []
print("Edges in the constructed MST")
for u, v, weight in result:
minimumCost += weight
print("%d -- %d == %d" % (u, v, weight))
nodes1.append(u)
nodes2.append(v)
weights_node.append(weight)
print("Minimum Spanning Tree", minimumCost)
return nodes1, nodes2, weights_node
```

To illustrate, the final form of the minimum spanning tree is presented in Figure 4.9 where the number of edges is equal to V-1 that was aforementioned in the paper. Also, it does not form any cycle. It is safe to conclude that GECS07 is mostly to be a suggested course by the minimum spanning tree to take in the second semester of the first year of a student.

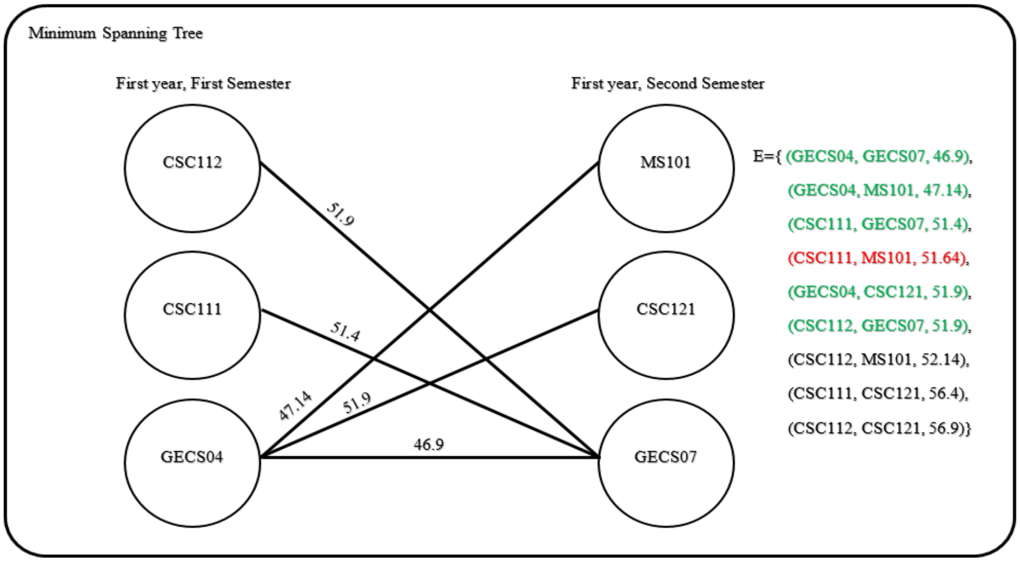


Figure 4.9 Final Form of the MST

4.2 Quick Design

Since the requirements are already identified, a design of the proposed application is being created. It is not a detailed design and covers the significant aspects of the system, which gives the user an idea of the system. This stage allows the user to have an overall view of what the proposed application offers and delivers.

4.2.1 Context Diagram

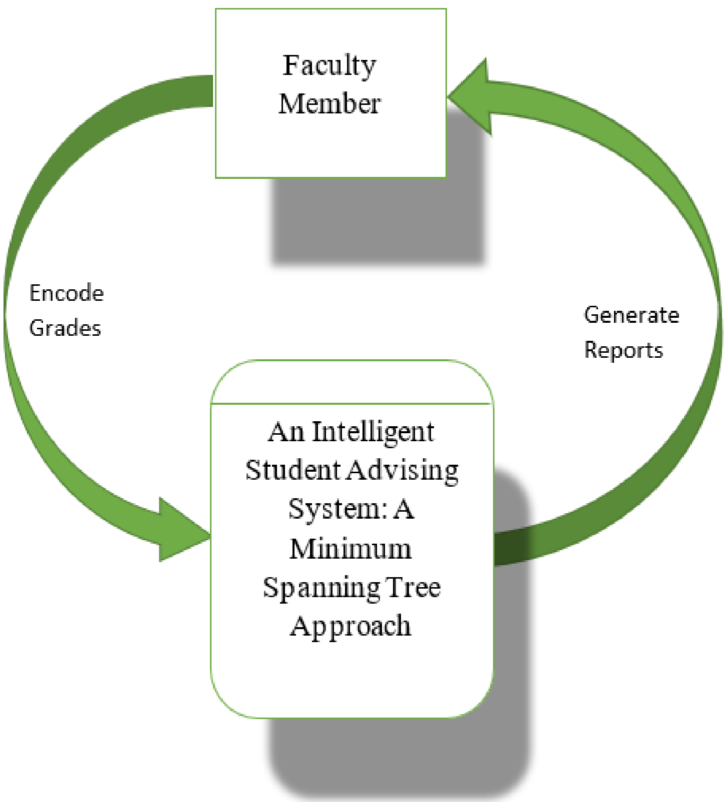


Figure 4.10 Context Diagram

The context diagram in Figure 4.10 shown above illustrates the authorized users input to the system and the expected output information to the users. The target user is the faculty member. The expected output of the proposed application are the list of subjects to be enrolled by the student, general weighted average based on inputted grades of the student and their academic status.

4.2.2 Data Flow Diagram of the Proposed System

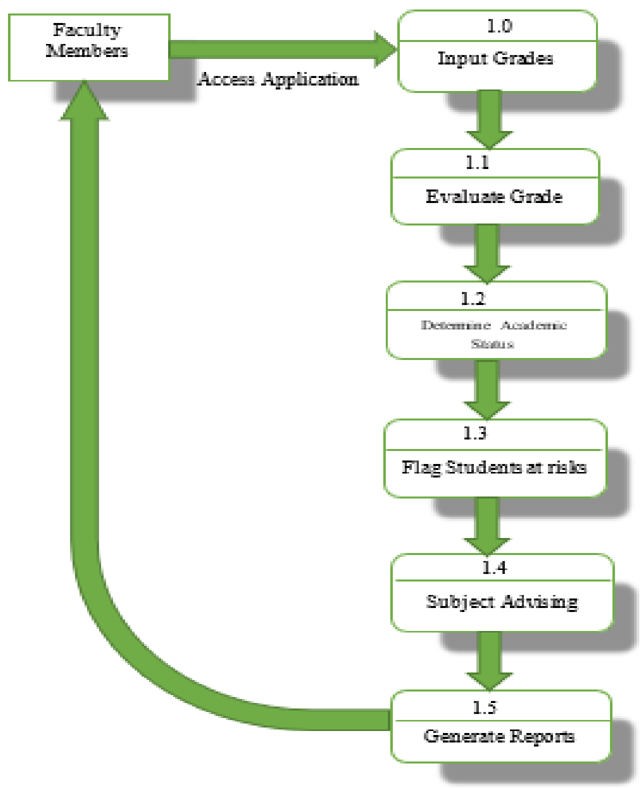


Figure 4.11 Data Flow Diagram

The researcher used the Data Flow Diagram, which is a dramatic representation of the information flow within a system that shows how information enters the system and leaves the system, what changes the information and where it is stored (Kendall, 2005).

Figure 4.11 illustrates the flow of the proposed system application. The faculty members will have to access the system to perform functions which include inputting of grades. Once done, the proposed application will compute the general weighted average, determine the academic status and provide list of subjects to be enrolled by the student

and flag students who are at risk of not graduating on time. The application is also capable of generating reports which contains the computed weighted average, academic status and list of subject for enrollment for the next semester.

4.2.3 Use Case

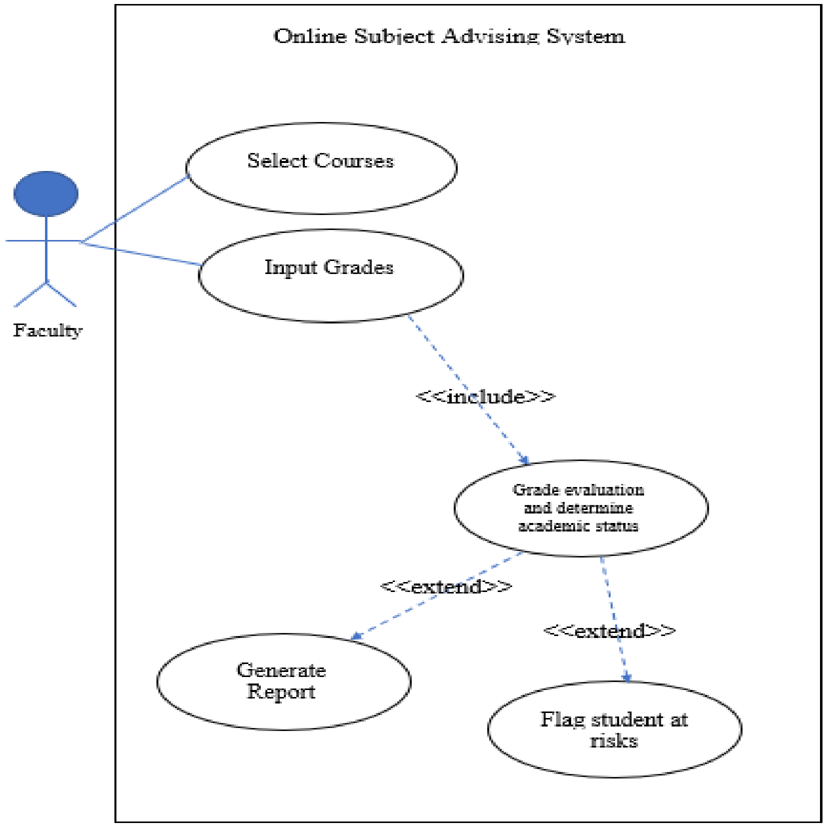


Diagram of the Proposed System

Figure 4.12 Proposed Use Case Diagram

The development of the proposed application does not solely depend on how the system works. It also depends on the workflow process that is being identified and needed to be implemented and followed. The components of the proposed application An Intelligent Student Advising System: A Minimum Spanning Tree Approach, is illustrated in Figure 4.12 with the use of Use Case Diagram. It describe its user, processes, and the relations between the system components that give the overall behavior of the application. Faculty members play an important role in the system. They are responsible

for encoding the students' grades and the system will compute the general weighted average automatically, determine the academic status, flag students who are at risk of not graduating on time and automate the list of subjects to be enrolled by the student.

4.2.4 Subject Advising Model

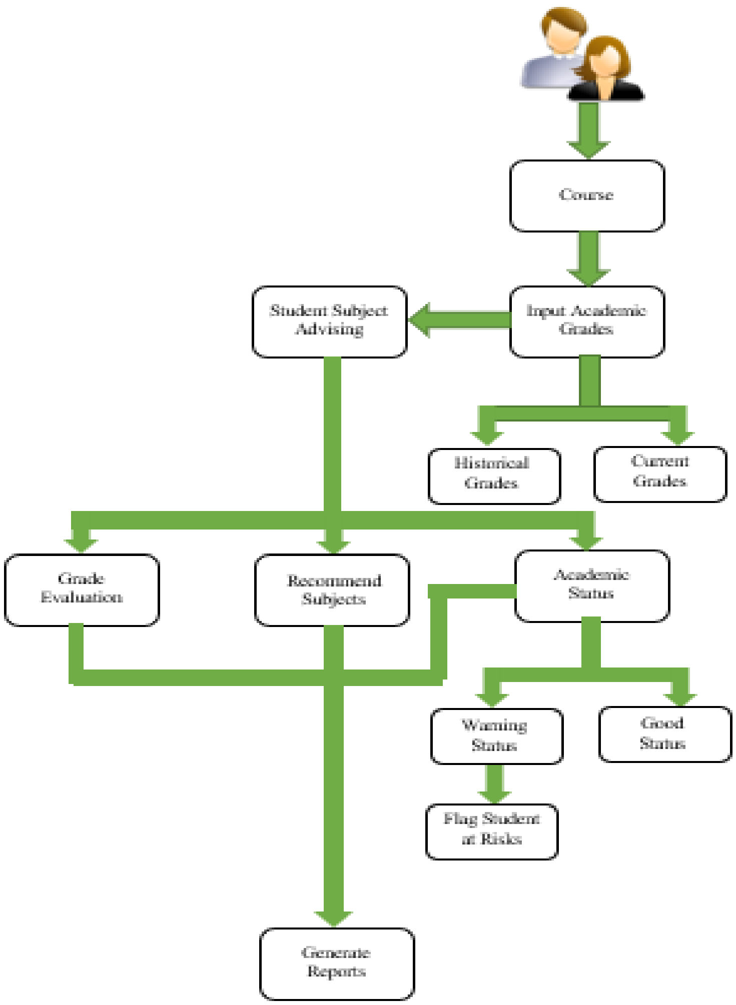


Figure 4.13 Subject Advising Model

Figure 4.13 shows the subject advising model. The user of the proposed application are the faculty members. The proposed application is projected to perform the

following processes: subject advising, status flagging, grade evaluation, and academic status determination of the students. The faculty member will then select a course, input historical grade or the student's previous semesters grade and the current semester grade. The proposed application will then compute for the general weighted average, determine academic status if it is warning or good status, and the system will be able to generate the list of recommended courses as well as the flagging to send an alert on the possibility of delay in finishing the program on time.

4.2.5 System Flowchart of the Proposed System

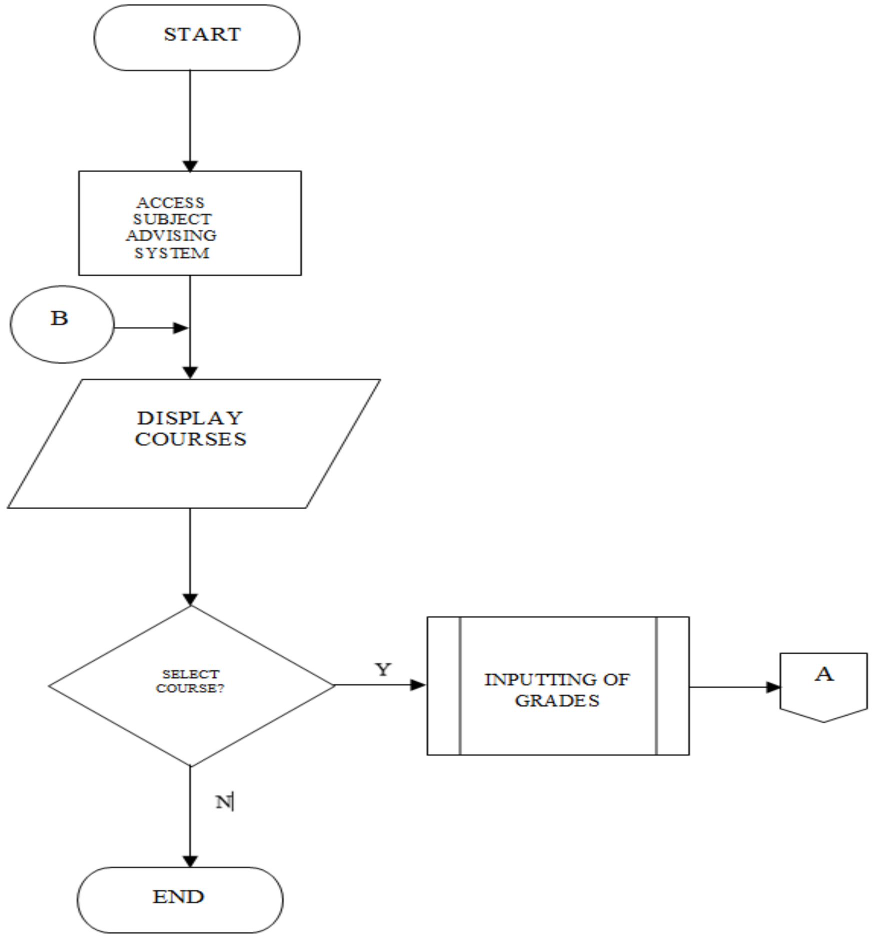


Figure 4.14 System Flowchart-Home Page

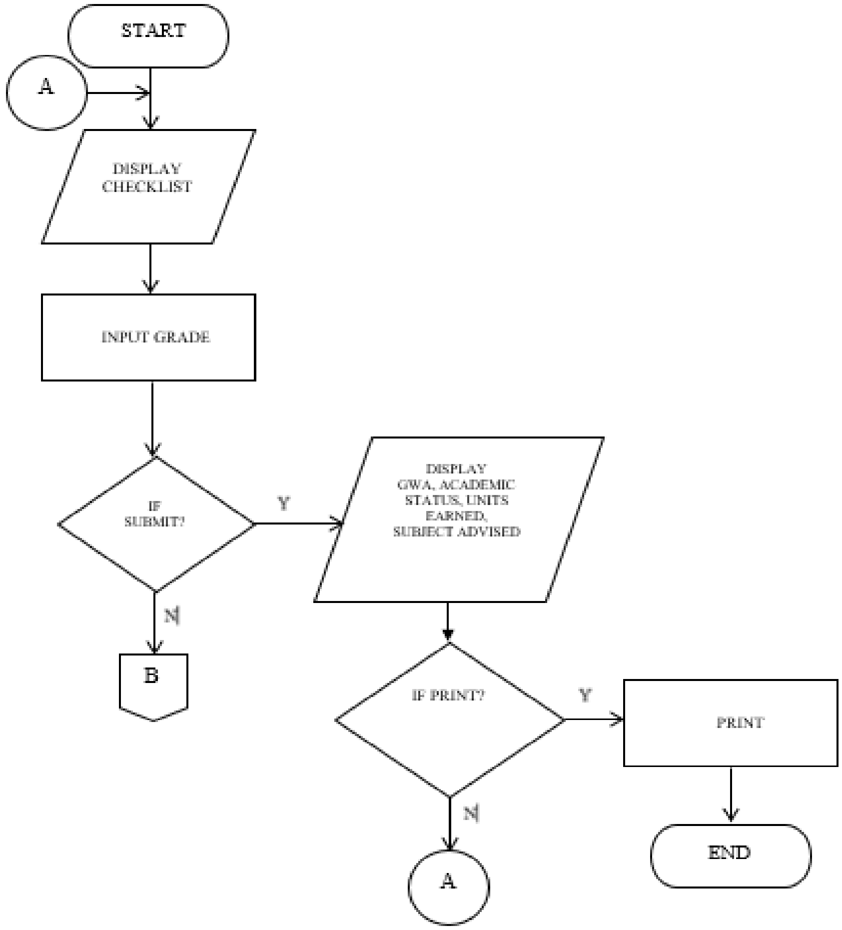


Figure 4.15 System Flowchart-Subject Advising Page

Figures 4.14 and 4.15 show the flow of the proposed system application. They illustrate how the application will be accessed by the user. To perform functions that include the input of grades, the user will have to access the system. Once completed, the proposed application will compute the general weighted average, determine the academic status and provide a list of subjects for students and flag students who are at risk of not graduating on time. The application can also generate reports containing the computed weighted average, the academic status and the list of subjects to be registered for the next semester.

4.3 Building Prototype

In the building prototype stage, the researcher developed the proposed application based on the designed context diagram, data flow diagram and system flowchart. The researcher identified the appropriate software and hardware needed in the development of the proposed system application to implement the concept presented in the design phase.

The proposed study An Intelligent Student Advising System: A Minimum Spanning Tree Approach can be implemented on any micro-computer configuration with the following capacities in order for the application to run smoothly: for the server, a minimum of 16GB RAM memory, a hard disk of at least 500GB and a minimum of 8

Core CPU; for the user of the application, 4GB RAM memory and a Dual Core CPU is recommended.

To develop the working prototype of the proposed system application, Hypertext Markup Language was used, which is a standardized system for tagging text files to achieve font, color, graphics, and hyperlink effects on the World Wide Web. Other applications that will be used is the Cascading Style Sheet (CSS). CSS is a language used for describing the look and formatting of a document written in a mark-up language. It is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting and reduce complexity and repetition in the structural content.

The researcher also utilized the Python programming language for the machine learning part of the application. Python is a general-purpose language that can do a set of complex machine learning tasks and enables the user to build prototypes quickly that would allow testing a product for machine learning purposes. The researcher utilized

PyCharm as the development environment for Python. After the model development, the

final model was deployed using Python Flask. It was made publicly available by uploading it on the Heroku web server.

4.4 User Evaluation

A survey questionnaire has been defined by Calderon & Gonzalez as simply a set of questions, which when answered properly by a required number of properly selected respondents, will supply the necessary information to complete a research study. The study utilizes a checklist type of survey questionnaire in which the respondents will be able to answer faster and easier at their convenience. The survey questionnaire utilized was based on the criteria evaluation on the system as guided by the ISO 9126-1. The survey questionnaire comprises the following criteria: Functionality; Efficiency; Reliability; Usability and Portability.

All faculty members were given equal chances to become one of the respondents. There are no specific qualifications considered in the selection of the respondents of this study aside from the fact that they are legitimate members of the teaching personnel of the Universidad de Manila.

Table 4.1

Verbal Interpretation Reference on Weighted Mean

Scale	Interpretation	Descriptive Equivalent
1	1.00-1.49	Needs Improvement (NI)
2	1.50-2.49	Fair (F)
3	2.50-3.49	Satisfactory(S)
4	3.50-4.49	Very Satisfactory (VS)
5	4.50-5.00	Excellent (E)

The responses on the questionnaire were graded using a 5-Point Likert scale. The rating has a qualitative description of: 1 – Needs Improvement, 2 – Fair, 3 – Satisfactory,
4 – Very Satisfactory, and 5 – Excellent. Data gathered through this questionnaire were statistically analyzed to come up with useful information in support of the study An Intelligent Student Advising System: A Minimum Spanning Tree Approach.

The proposed application is presented to the intended user for thorough evaluation of the prototype. Comments and suggestions were collected from them. Most of the comments are focused on the graphical user interface and the applications functionality. Most notable among these comments is that it does not correctly compute the general weighted average of the students. Another comment points out that the application lacks instructional messages on how to use it.

4.5 Refining Prototype

In this stage, the end user evaluated the prototype. Dissatisfaction with the prototype at this level resulted in revision based on the given requirements. The new prototype was re-evaluated and the process continued until such time that the requirements identified by the end-user were met. Revisions were done based on the user's comments and suggestions during the evaluation of the developed application.

The application was revised and enhanced, applying all the needed modifications. Several testing sessions were done by the user until all the requirements specified are met. When the user's satisfaction was finally met, a final prototype was developed.

4.6 Engineer Product

The last stage of this approach concludes with the confirmation and approval of the application by the end-users. This is also referred to as the user acceptance phase. It is also in this phase that the researcher was able to appraise the overall performance of the

final system, using the predetermined indices or indicators such as functionality, efficiency, reliability, usability and portability.

Prototyping has been proven beneficial since it ensures quality system delivery, prompt problem diagnosis, evident end-user participation, fulfillment of end-user requirement, and reduced utilization cost.

Chapter Five

RESULTS AND DISCUSSION

This section presents, analyzes and interprets the results of the study in developing an intelligent student advising system for Universidad de Manila. It also presents the result of the survey among 92 faculty members of the university, purposely to evaluate the proposed intelligent student advising system in terms of accuracy. Due to the very small population of the respondents, Simple Random Sampling was considered the best suited sampling technique and employed the use of Sloven's Formula to generate the enough data samples needed in determining the accuracy of the proposed application entitled, An Intelligent Student Advising System: A Minimum Spanning Tree Approach.

5.1 Functionality of the Newly Developed System

5.1.1 Subject Advising

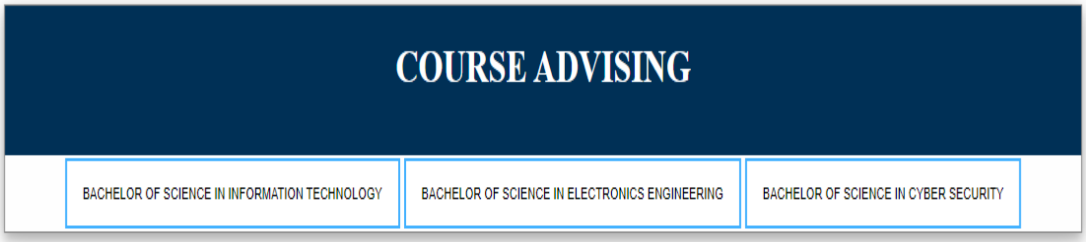


Figure 5.1 Main Screen of Student Advising

Interaction between the user and the system is simply designed through a friendly user interface. Figure 5.1 shows the main screen of the system application where the user can choose among the three courses which are the Bachelor of Science in Information Technology, Bachelor of Science in Electronics Engineering and Bachelor of Science in Criminology. Upon clicking on specific course, the user will be directed to the Input Grade Page.

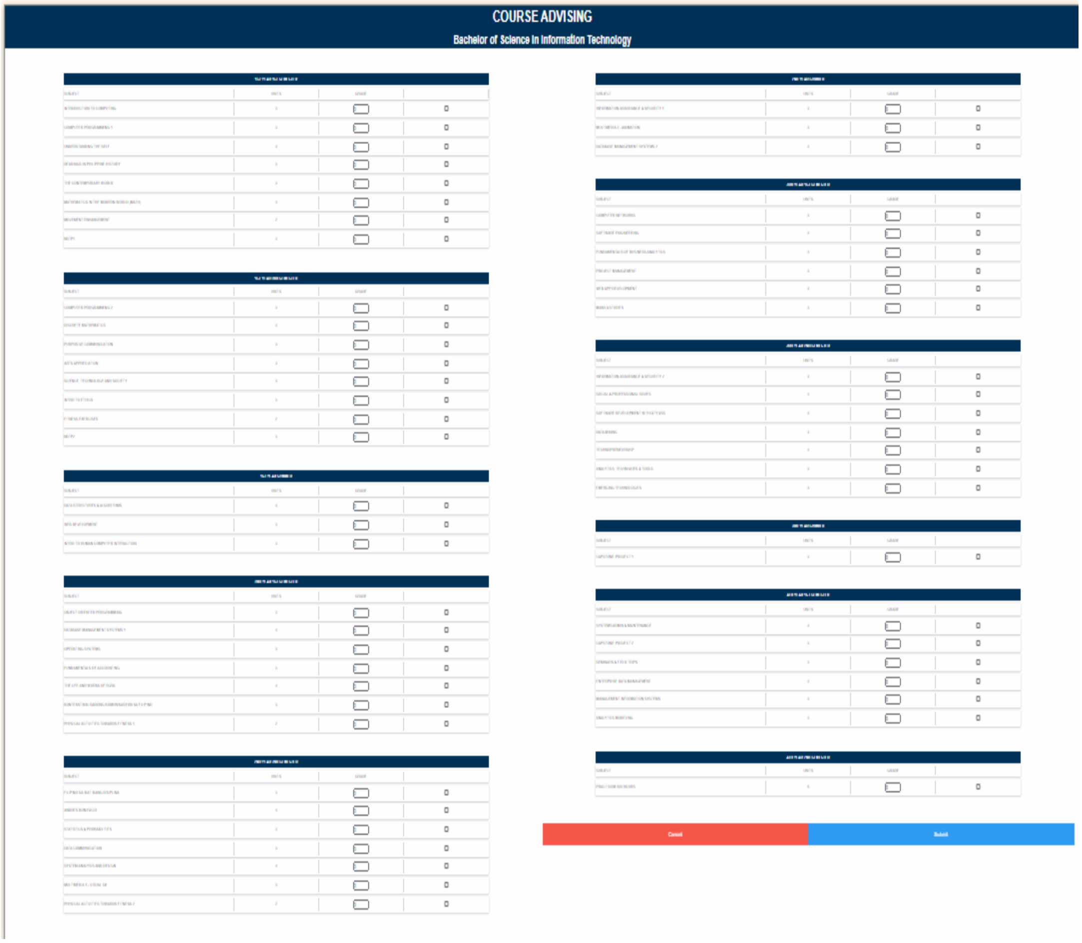
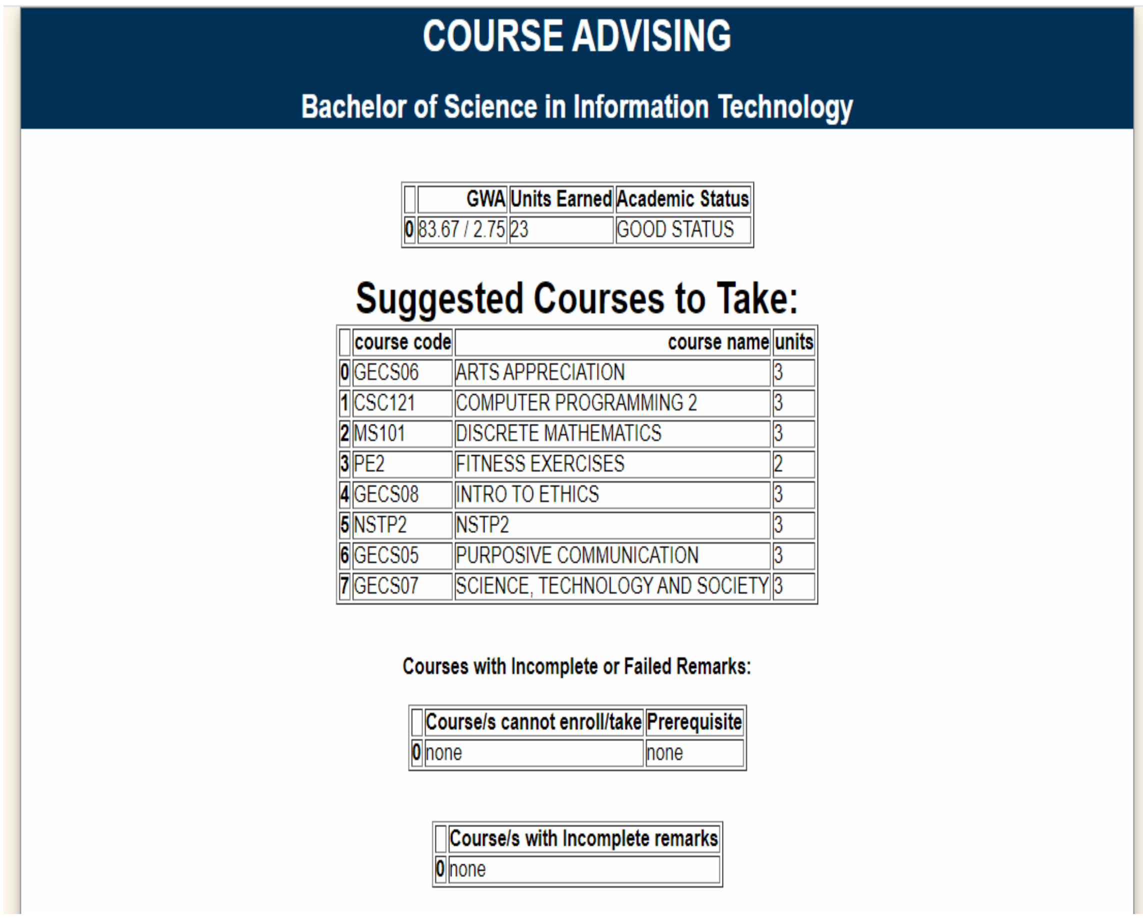


Figure 5.2 Inputting of Grades

Figure 5.2, shows the interface for Input Grades that allows the user to input grades of the students for a specific subject. There are two options to choose from: the Cancel button and the Submit button. Once the user is done inputting the student's grades, he needs to click on the submit button for him to be directed to the subject advising page. If the user wants to go back to the previous page and choose another course the user may click on the Cancel button.



COURSE ADVISING

Bachelor of Science in Information Technology

	GWA	Units Earned	Academic Status
0	83.67 / 2.75	23	GOOD STATUS

Suggested Courses to Take:

	course code	course name	units
0	GECS06	ARTS APPRECIATION	3
1	CSC121	COMPUTER PROGRAMMING 2	3
2	MS101	DISCRETE MATHEMATICS	3
3	PE2	FITNESS EXERCISES	2
4	GECS08	INTRO TO ETHICS	3
5	NSTP2	NSTP2	3
6	GECS05	PURPOSIVE COMMUNICATION	3
7	GECS07	SCIENCE, TECHNOLOGY AND SOCIETY	3

Courses with Incomplete or Failed Remarks:

	Course/s cannot enroll/take	Prerequisite
0	none	none

	Course/s with Incomplete remarks
0	none

Figure 5.3 Recommended Courses by the System Application

Upon clicking the submit button, the user will be directed to the Recommended Course page. In Figure 5.3, shown are the recommended courses based on the grades entered by the user. The application is capable of recommending subjects that can be enrolled commensurate to the grades of the students. It can be noted that if a student fails a subject with prerequisite, he will not be able to enroll in the said subject.

5.1.2 Flagging of Students

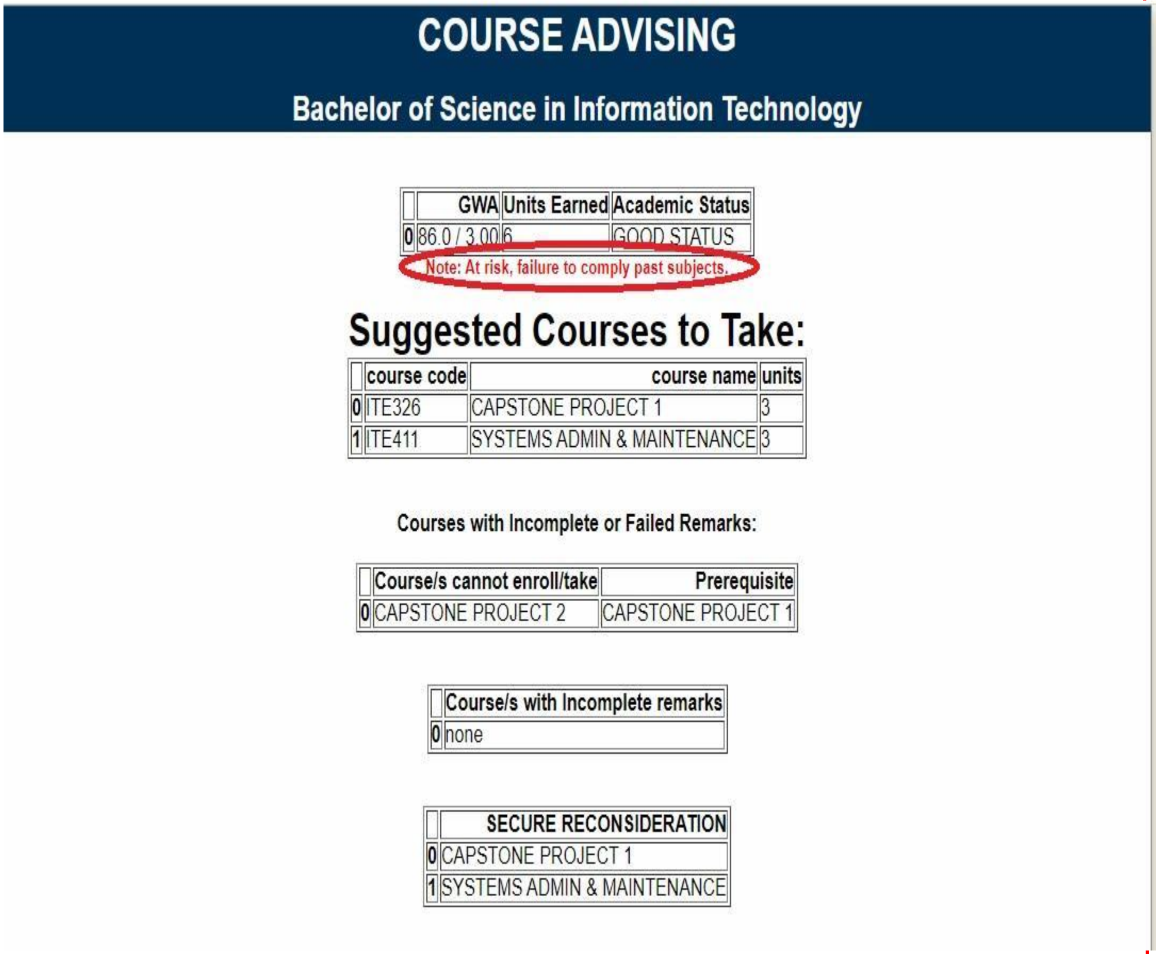


Figure 5.4 Student Flagging

Figure 5.4, it shows that the application is capable of tagging students who are at risk of not graduating on time. Once the student acquires a failing grade, he will be automatically flagged or marked.

5.1.3 Determine Student Academic Status

The application will also display the computed average and the student's academic status. The academic status is either Good Standing or Warning Status. If the student has no failing grade, he will be marked as Good Status while if a student has a failing grade, he will be marked with a Warning Status and be flagged as not graduating on time. The system application is capable of generating reports simply by clicking the Print button.

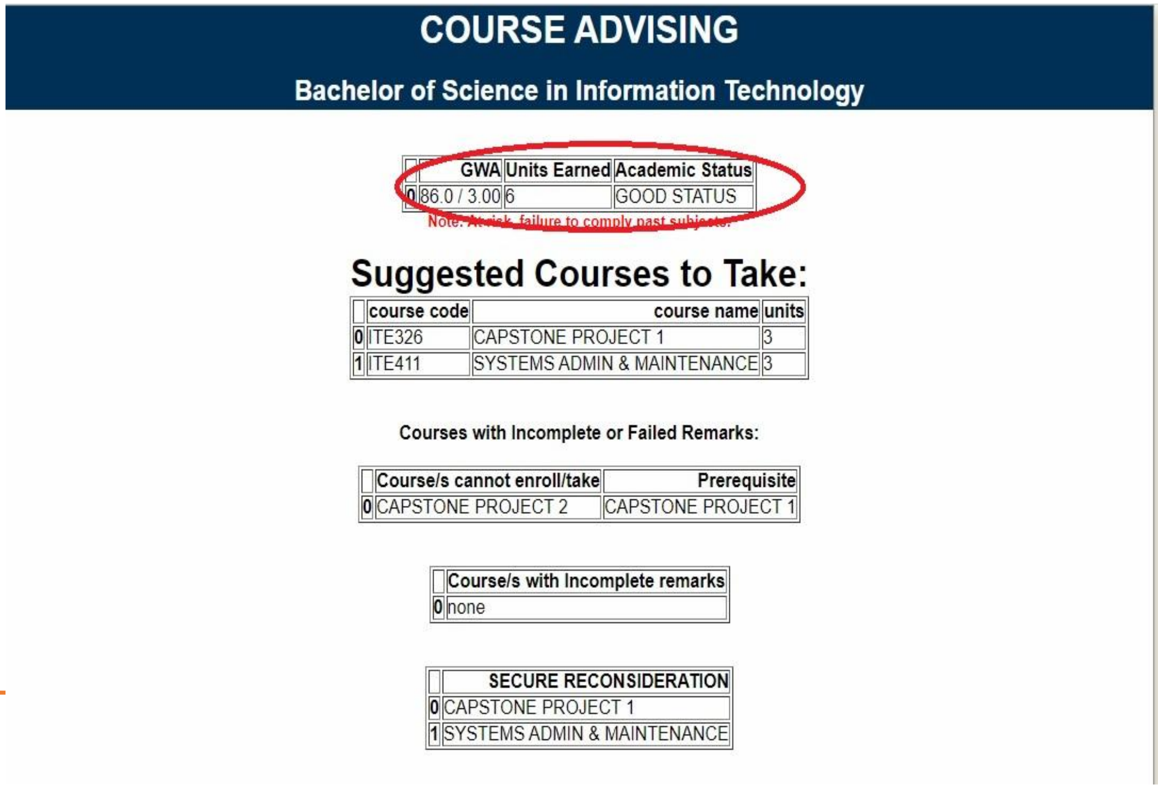


Figure 5.5 Student Academic Status

5.1.4 Evaluation of the Newly Developed Student Advising System

After testing and administering the user's evaluation survey, the researcher gathered the results, tallied, analyzed and interpreted the results. Table 5.1 presents the verbal interpretation reference on weighted mean with the following Likert's scale ranges.

Table 5.1

Verbal Interpretation Reference on Weighted Mean

Scale	Interpretation	Descriptive Equivalent
1	1.00-1.49	Needs Improvement (NI)
2	1.50-2.49	Fair (F)
3	2.50-3.49	Satisfactory(S)
4	3.50-4.49	Very Satisfactory (VS)
5	4.50-5.00	Excellent (E)

5.1.5 Functionality of the Newly Developed Application as Evaluated by the Faculty Members

Table 5.2 presents the faculty members responses on the functionality of the proposed application. Techtarget (2009) defined functionality as the sum or any aspect of what a product, such as a software application or computing device can do for a user. It is how the characteristics of the system work to provide a desired result for the users and its ability to perform the job it was designed for.

Table 5.2

The Weighted Mean on the Evaluation of Functionality

FUNCTIONALITY	Weighted Mean	Verbal interpretation
1. Completeness of the application	4.82	Excellent
2. Functional Appropriateness	4.80	Excellent
3.Functional Correctness	4.89	Excellent
General Weighted Mean	**4.84**	**Excellent**

Table 5.2 shows the weighted mean on the evaluation of functionality, the first criteria on user's evaluation. The functionality consists of 3 indicators in which all of these obtained an "excellent" rating. Item no. 3, *Functional Correctness*, got the highest weighted mean of 4.89. Furthermore, the general weighted mean on this criterion is 4.84 and is interpreted as "Excellent", an indication of its serviceability in the system. Results have shown that the system serves its original purpose of subject advising. The relatively higher evaluation of the faculty members on item no. 3 which is Functional Correctness indicates that the application is very accurate in terms of providing the correct results or outputs with a degree of precision.

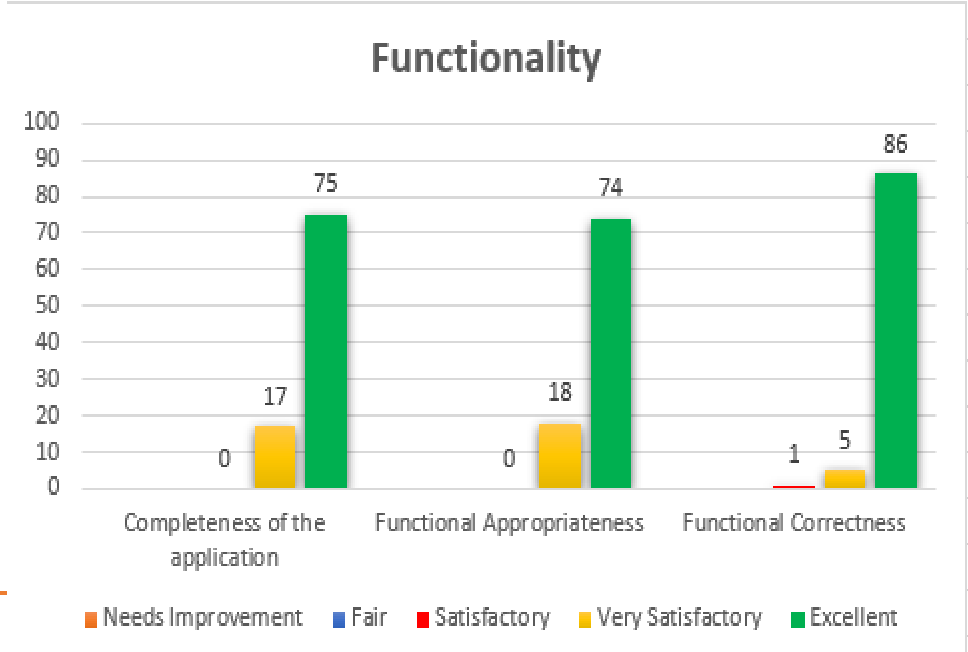


Figure 5.6 Qualitative Evaluation of Functionality

Figure 5.6 emphasizes the qualitative evaluation on functionality. This was evaluated based on three sub-criteria: the completeness of the application, functional appropriateness and the functional correctness. It is viewed from the figure that most of the respondents provided an evaluation rating of 5 to all sub-criteria which is "Excellent"

and few from "Very Satisfactory" rating. Furthermore, these can be deduced from the results that the system occurred with highly functional requirements or criteria.

5.1.5.1 Efficiency of the Intelligent Student Advising System as Evaluated by the Faculty Members

Efficiency involves reducing the amount of unnecessary resources, including personal time and energy, that are required to generate a given output.

Table 5.3
The Weighted Mean on the Evaluation of Efficiency

EFFICIENCY	Weighted Mean	Verbal interpretation
1. Time Behavior	4.49	Very Satisfactory
2. Resource Utilization	4.49	Very Satisfactory
General Weighted Mean	**4.49**	**Very Satisfactory**

The weighted mean on the evaluation of efficiency is presented in Table 5.3. It is noted from the table that the two indicators have the same weighted mean (4.49), which is interpreted as "Very Satisfactory". This can be associated with the fact that there were no I/O related errors which are system dependent that have been detected during the testing process. The response time is extremely faster than the existing manual subject advising.

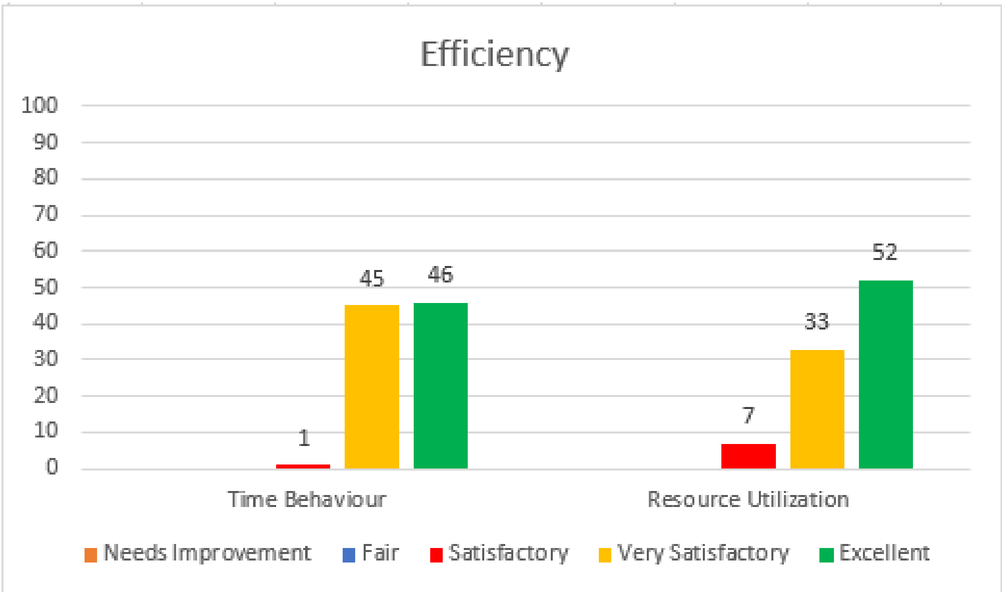


Figure 5.7 Qualitative Evaluation of Efficiency

A qualitative evaluation on efficiency is reflected in Figure 5.7. The second criterion to consider is the efficiency of the system, which comprises two sub-criteria: the time behavior and the resource utilization. Viewed from the figure, it provided mostly with two evaluation ratings, that is, 4 (Very Satisfactory) and 5 (Excellent) and with a very small value of "Satisfactory" rating. Combining all these three ratings, it resulted in a very satisfactory requirement of the system.

The researcher used another method to measure the efficacy and the overall performance of the application by means of numerical analysis of figures gathered by measuring the time or speed of the completion of the process.

Manual Process vs Proposed System Speed Evaluation		
Instance	**Manual Process**	**Proposed System**

To get values and the average of all measures, the prototype was subjected to testing for several times. The series of tests done resulted to the following:

1	7:10	1:50
2	9:10	2:04
3	7:23	2:00
4	8:37	2:56
5	7:40	4:45
6	8:12	3:20
7	9:30	6:10
8	9:25	2:21
9	10:38	3:42
10	9:28	3:50
Average Processing Time	**8:43 mins**	**3:17 mins**

Table 5.4

Benchmarking Analysis (Speed)

Ten instances have been simulated, checked, and time-stamped to test for the pace of the processing of evaluating students through different methods; manual process and automated process. Manual processing takes a longer time since a student needs to update the checklist before evaluation takes place. Once done, the faculty member has to compute for the general weighted average of the student and determine his academic status. Thereafter, the faculty member advises the students, especially the irregular ones, on what subject to enroll. In doing so, the faculty needs to double-check the prerequisites of subjects if there is any. The faculty member will then make a recommendation on what subjects the student could take.

Subject advising through manual process may take time since it highly depends on the case of every student. A student who has backlog or failed subjects will definitely take longer to advice. Contrary to the automated process, the assigned faculty member only needs to input the grade of the student in application then the system will automatically compute for the general weighted average, determine the academic status, flag the students for the possibility of not being able to graduate on time, and recommend the list of subjects for enrolment. The reason for processing time of the automated process to extend is connection difficulty, just like what happened during its evaluation where the processing time took 6.10 minutes.

.

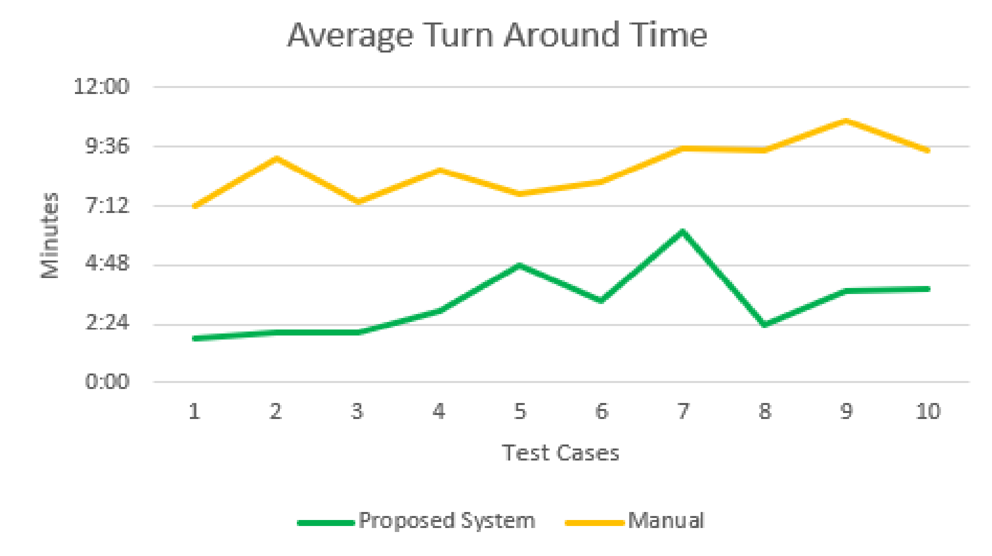


Figure 5.8 Speed Evaluation

Figure 5.8 shows the difference between the manual and automated student evaluation and subject advising process. There is a drastic difference between the manual processes with the automated process. The manual process has an average processing of 8.43 minutes while the automated process has 3. 17 minutes average processing time. This indicates that the response time of the automated student subject advising system is extremely faster than the existing manual subject advising.

5.1.5.2 Reliability of the Intelligent Student Advising System as Evaluated by the Faculty Members

Reliability is defined as the ability of an equipment, machine or system to consistently perform its intended or required function on demand without degradation or failure (Business dictionary.com, 2009). A reliable system provides consistent results after several viewing.

Table 5.5

The Weighted Mean on the Evaluation of Reliability

RELIABILITY	Weighted Mean	Verbal Interpretation
1. Availability	4.95	Excellent
2. Fault Tolerance	4.38	Very Satisfactory
3. Recoverability	4.32	Very Satisfactory
General Weighted Mean	**4.55**	**Excellent**

Table 5.5 shows the weighted mean on the evaluation of reliability. From the table, it is revealed that among the three indicators, one is interpreted as "Excellent"

with a weighted mean of 4.95, that is, item number 1, the *Availability*. This means that the

application is operational and accessible when required for use. A very satisfactory rating was also obtained with respect to recoverability with a weighted mean of 4.32. The 13 respondents who gave satisfactory responses failure during the evaluation of the application. In such cases, they need to re-input the grades of the students, this problem will be eliminated once the application is integrated in the current student information system that is being utilized by the university. The same findings can be noted in terms of fault tolerance having a weighted mean of 4.38. The very satisfactory evaluation of the faculty members suggests that user input can be verified by the application. It is also seen from the table that the general weighted mean on reliability is 4.55, which is interpreted as "Excellent". This only proves that the application is capable of providing consistent results after several viewing and that the system is capable of performing the required tasks based on its objective.

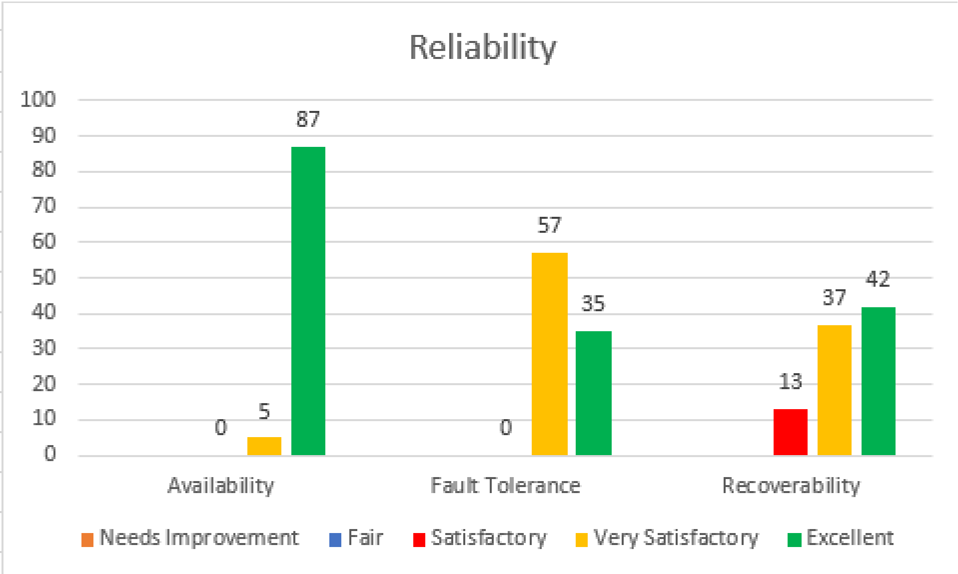


Figure 5.9 Qualitative Evaluation on Reliability

Figure 5.9 presents the graphical presentation of the qualitative evaluation on reliability. The criterion reliability has three sub-criteria such as availability, fault tolerance, and recoverability. This is one vital feature of the overall performance of a system which provides consistent performance. It is noted from the figure that the sub-criterion availability has supplied the greatest number of excellent ratings. And it is

viewed also that the sub-criterion fault tolerance has the most number of very satisfactory ratings. However, the fault tolerance and recoverability resulted in a very satisfactory mark.

5.1.5.3 Usability of the Intelligent Student Advising System as Evaluated by the Faculty Members

The effectiveness, efficiency, and user satisfaction all count under the usability of the system. It practically refers to the quality of the user's experience while using systems in the form of websites, software, devices, or applications.

Table 5.6

The Weighted Mean on the Evaluation of Usability

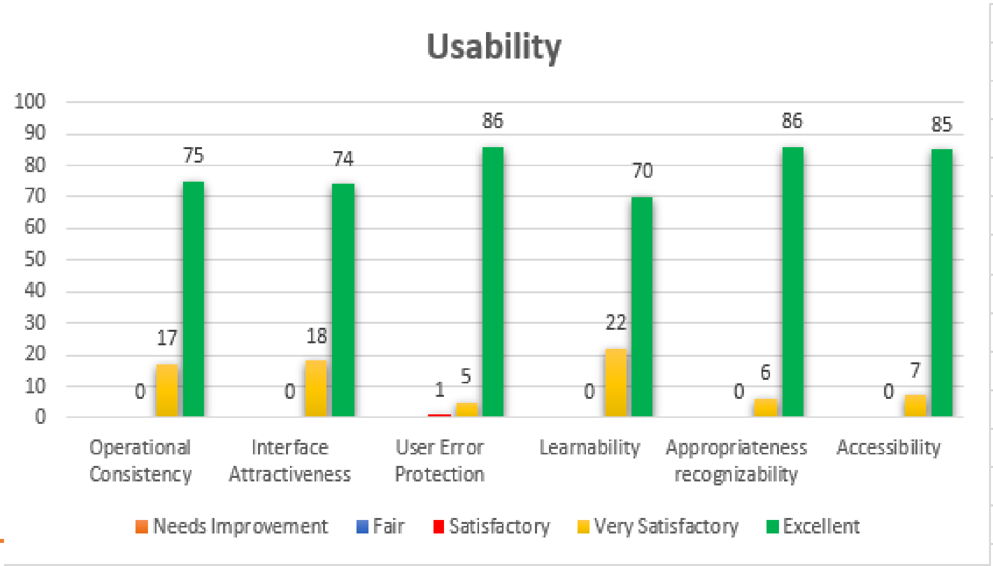
Usability	Weighted Mean	Verbal Interpretation
1. Operational Consistency	4.97	Excellent
2. Interface Attractiveness	4.85	Excellent
3. User Error Protection	4.90	Excellent
4. Learnability	4.83	Excellent
5. Appropriateness recognizability	4.87	Excellent

6. Accessibility	4.86	Excellent
General Weighted Mean	**4.88**	**Excellent**

The weighted mean on the evaluation of usability is revealed in Table 5.6 Similar to functionality, all of the items obtained an interpretation of "Excellent" which is also the interpretation for the entire usability with a general weighted mean of 4.88. This indicates that the application is found to be user friendly, visually appealing and easy to learn. Among the six indicators, item 1 (operational consistency) shows the highest weighted rating of 4.97 which is almost perfect. This implies that the developed application has attributes that make it easy to operate and control.

Figure 5.10 is the graphical presentation of the qualitative evaluation on usability. Usability can be described as the capacity of a system to provide a condition for its users to perform the tasks safely, effectively, and efficiently. This includes the six sub-criteria of operations consistency, interface attractiveness, user error protection, learnability, appropriateness recognizability, and accessibility.

igure 5.10 Qualitative Eval



uation on Usability

It is revealed from the figure that the majority of the six sub-criteria furnished the excellent rating with a very small amount of very satisfactory. This may prove that the usability in the system is vital in a user's experience when interacting with the system.

To ensure the flexibility and adaptability of the application, the web app was made to be responsive and can adjust to the user's behavior and environment such as the user's device screen size, platform/browser (Opera, Google Chrome, Mozilla Firefox, etc.) and its orientation. Since the target users of the application would be the faculty, the overall interface was made to be simple so the users can have ease of learning and navigate the important features of the program.

5.1.5.4 Portability of the Intelligent Student Advising System as Evaluated by the Faculty Members

Portability is the ability to transfer applications from one computer device to another. It refers to application software that can be recompiled for a particular platform or portable for different platforms.

Table 5.7

The Weighted Mean on the Evaluation of Portability

PORTABILITY	Weighted Mean	Verbal Interpretation
1. System Adaptability	4.95	Excellent
2. User Adaptability	4.96	Excellent

3. Ease of Installation	4.98	Excellent
General Weighted Mean	**4.96**	**Excellent**

The weighted mean on the evaluation of portability, the last criteria is posted in Table 5.7. Portability refers to ease of installation, user adaptability and system adaptability. These features help the user run and install the developed application in any platform. It also refers to the variability of devices such as desktop computers, tablet PC and mobile phones. It is shown here that the three indicators obtained an "Excellent" interpretation in which each reveals an almost perfect score, especially item number 3, *Ease of Installation*. And the general weighted mean, as seen from the table, obtained also an almost perfect score of 4.96 which is interpreted as "Excellent". This can be due to that fact that the application is web-based so no installation is necessary and the system is readily available via web browser. The interface is also designed simply so the users can easily learn how to use the application.

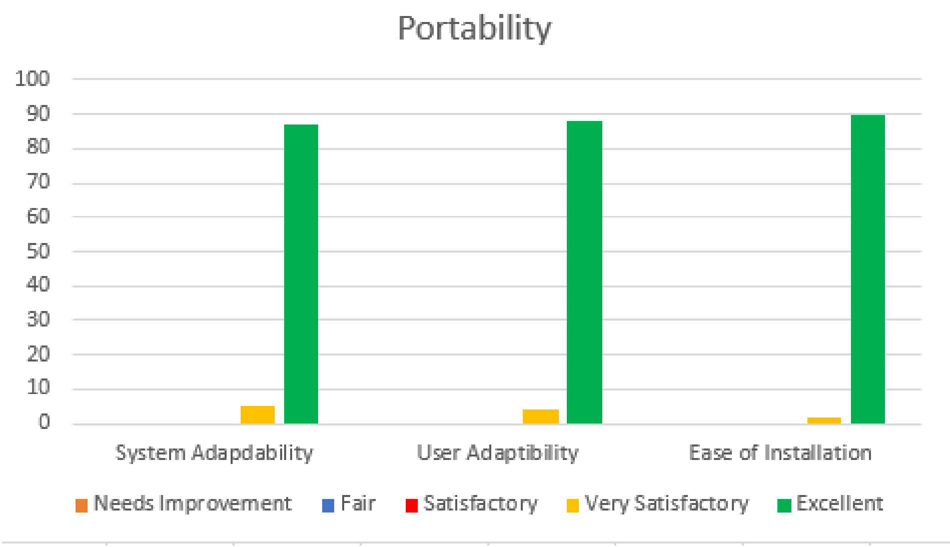


Figure5.11 Qualitative Evaluation on Portability

One important criterion in the system is portability. The qualitative evaluation on portability is shown in Figure 5.11. From the figure, all of the criteria have an excellent rating which is an indicative of easiness and adaptability of the system.

5.1.5.5 Summary of User Evaluation on the Five Criteria

Table 5.8

Summary of User Evaluation

Criteria	General Weighted Mean	Verbal interpretation
1. Functionality	**4.86**	**Excellent**
Completeness of the application	4.82	Excellent
Functional Appropriateness	4.80	Excellent
Functional Correctness	4.89	Excellent
2. Efficiency	**4.49**	**Very Satisfactory**
Time Behavior	4.49	Very Satisfactory
Resource Utilization	4.49	Very Satisfactory
3. Reliability	**4.55**	**Excellent**
Availability	4.95	Excellent

Fault Tolerance	4.38	Very Satisfactory
Recoverability	4.32	Very Satisfactory
4. Usability	**4.88**	**Excellent**
Operational Consistency	4.97	Excellent
Interface Attractiveness	4.85	Excellent
User Error Protection	4.90	Excellent
Learnability	4.83	Excellent
Appropriateness Recognizability	4.87	Excellent
Accessibility	4.86	Excellent
5. Portability	**4.96**	**Excellent**
System Adaptability	4.95	Excellent
User Adaptability	4.96	Excellent
Ease of Installation	4.98	Excellent
Overall Grand Weighted Mean	**4.75**	**Excellent**

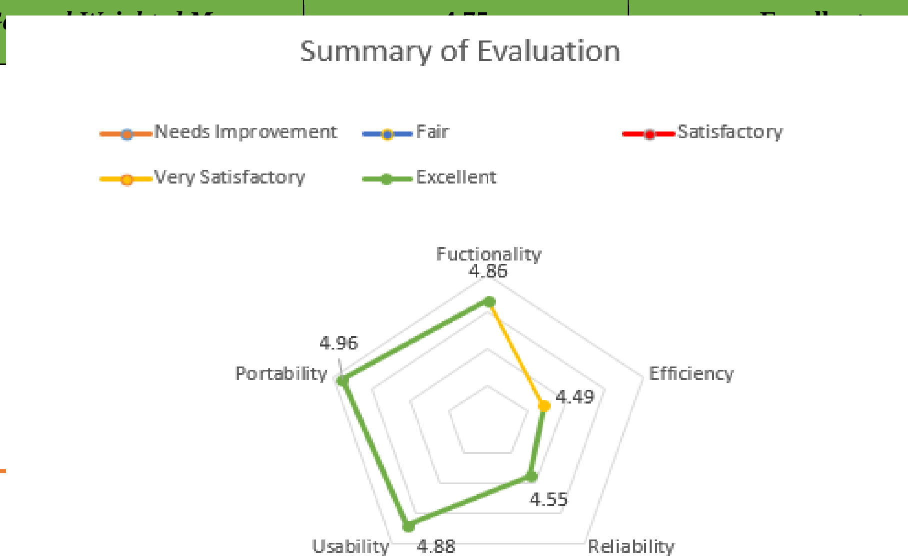


Figure 5.12 Evaluation Summary

Table 5.8 and Figure 5.12 presents the summary of user evaluation on the five criteria. The overall evaluation score of the system is 4.75 and is interpreted as “Excellent”, which indicates the serviceability of the system. The overall general weighted mean is the average of the five-criterion evaluation on the system. It is revealed from the table that the lowest mean point was obtained by the criterion efficiency (4.49). This may be due to the fact that it takes time to input the grades of a student from the previous semester to the current semester. To solve the problem, the application should be integrated into the current student information system of the university. Through this, the time spent from inputting the grades of a student will be minimized. Another factor may be due to the fact that the application is using a free web hosting site which has limited processing power.

On the other hand, portability criterion got the highest evaluation rating with a general weighted mean of 4.96. This implies that though the respondent believe that the intelligent student advising system is efficient, its portability and ability to adapt in any platform or any working environment is better appreciated

Chapter Six

CONCLUSIONS AND RECOMMENDATIONS

This section contains the discussions of the conclusions and researcher recommendations based on the results of the study.

6.1 Conclusions

Based on the stated objectives of the study, the researcher concludes the following:

1. The excellent overall evaluation of the faculty members on the functionality, efficiency, reliability, usability and portability is based on how the respondents perceive the outputs of the intelligent student advising system. This means that presently, the application was able to deliver what is expected of it by the end user.

2. The tools used in the development of the Student Advising System: A Minimum Spanning Tree Approach made a major contribution to the overall objective of designing this application. The developed Student Advising System has a lot of remarkable advantages compared to the manual processes of the

subject advising system. This conclusion is based on the testing for accuracy and speed testing. There is a drastic difference in terms of providing accurate results, ease of use and speed of processing with the benchmarking method. The application established was proved accurate and this was justified by the findings of the assessment performed. The developed application, Intelligent Student Advising System: A Minimum Spanning Tree Approach is designed to automate the subject advising system utilized by the university.

6.2 Recommendations

In a study entitled, Building a Smart Academic Advising System Using Association Rule Mining authored by Raed Shatnawi et. al., in an academic

environment, student advising is considered a paramount activity for both advisors and students to improve the academic performance of students. In universities of large numbers of students, advising is a time-consuming activity that may take a considerable effort for advisors and university administration in guiding students to complete their registration successfully and efficiently. Current systems are traditional and depend greatly on the effort of the advisor to find the best selection of courses to improve the students' performance. There is a need for a smart system that can advise a large number of students every semester. In this paper, a smart system that uses association rule mining to help both students and advisors in selecting and prioritizing courses was proposed. The system helped the students improve their performance by suggesting courses that meet their current needs and at the same time improve their academic performance. The system uses association rule mining to find associations between courses that have been registered by students in many previous semesters. The system successfully generates a list of association rules that guide a particular student to select courses registered by similar students.

Therefore, this study recommends the use of an intelligent subject advising since it can provide the following advantages:

1. The developed application can recommend the list of subjects to be enrolled by the students.

2. The developed application has the ability to identify students who are at risk of not graduation on time.

3. It has the capability to determine the academic status of the students and the application is accurate in providing results.

Further development and enhancement of the system is thereby recommended to future researchers, especially to include the following:

1. Integration to the existing student information system utilized by the university so that filed grades of the students will be systematically retrieved from the existing database.

2. Include as part of the system the other colleges and the Graduate School to simulate the advising process.

3. Implementation of the same algorithm and model in a mobile application. This way, portability and accessibility options can also be improved.

LIST OF REFERENCES

Abalorio,Cristopher and dela Cerna,Monalee. (2018). *Course Evaluation Generator (Ceg): An Automated Academic Advising System with Optical Character Recognition*

Asogbon, Mojisola G.; Samuel, Oluwarotimi W.; Omisore, Mumini O., and Ojokoh, Bolanle A.. (2016). *A Multi-class Support Vector Machine Approach for Students Academic Performance Prediction*

Bae, Rodel Emille T.; Arboleda, Edwin R.; Andilab, Adonis, and Dellosa,Rhowel M..(2019). *Implementation Of Template Matching, Fuzzy Logic And K Nearest Neighbor Classifier On Philippine Banknote Recognition System*

Beuret, M., Billot, N., Cambrésy, L., Eden, D. J., Elia, D., Molinari, S., Pezzuto S., and Schisano, E. (2017). *Source clustering in the Hi-GAL Survey Determined Using A Minimum Spanning Tree Method*

Cao,Jianfang; Wang,Min; Li, Yanfei and Zhang, Qi. (2019). *Improved Support Vector Machine Classification Algorithm Based on Adaptive Feature Weight Updating in the Hadoop Cluster Environment*

Carvajal, Thaddeus M.; Viacrusis, Katherine M.; Hernandez,Lara Fides T.; Ho,Howell T.; Amalin, Divina M., and Watanabe, Kozo. (2018). *Machine learning methods reveal the temporal pattern of dengue incidence using meteorological factors in metropolitan Manila, Philippines*

Catapang, Jasper Kyle. (2019). *k-Nearest Neighbor Optimization via Randomized Hyperstructure Convex Hull*

Chen,Jianguo; Li,Zhuo,Kenli; Bilal, Tang,Kashif;Yu, Shui; Weng ,Chuliang and Li ,

Keqin (2018). *A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment*

Choudhury, Ambika. (2019). *Basics of Ensemble Learning In Classification Techniques Explained*

Co, Jan Miles and Fernandez, Proceso. (2017). *Time-Series Link Prediction Using Support Vector Machines*

Daramola,Olawande; Emebo,Onyeka; Afolabi,Ibukun, and Ayo,Charles. (2014). *Implementation of an Intelligent Course Advisory Expert System Cased-Based Course Advisory Expert System*

Ezz, Mohamed M., (2015). *Advisory System for Student Enrollment in University Based on Variety of Machine Learning Algorithms*

Faizaan Shalik, Srikanth Benzawada, Neena Goveas. (2015). *CySpanningTree: Minimal Spanning Tree computation in Cytoscape*

Galupino, Joenel G. and Dungca, Jonathan R... (2018) .*Quezon City Soil Profile Reference*

Ganeshan, Kathiravelu and Li, Xiaosong (2015). *An Intelligent Student Advising System Using Collaborative Filtering*

Grewal DS and Kaur, (2016). *Developing an Intelligent Recommendation System for Course Selection for Graduate Courses*

Gaunther, Nick and Schonlau, Matthias. (2016). *Support vector machines*

Handayani, Irma. (2019). *Application of K-Nearest Neighbor Algorithm on Classification of Disk Hernia and Spondylolisthesis in Vertebral Column*

Lutins, Evan (2017). *Ensemble Methods in Machine Learning: What are they and Why Use Them?*

Kadambande, Ankita; Thakur, Snehal; Mohol, Akshata; Prof Ingole, A.M... (2017). *Predict Student Performance by Utilizing Data Mining Technique and Support Vector Machine*

Liaw, Andy and Wiener, Matthew (2001). *Classification and Regression by RandomForest*.

Maladkar , Kishan (2018). *Solving the Titanic ML Survival Problem Using Random Forest vs Neural Networks on Tensorflow, Which One is Better?*

Malicdem, Alvin R. and Fernandez, Proceso L... (2014).*Kernel-based Support Vector Machine for Rice Blast Disease Prediction in Northern Philippines*

Ms.Alfarsi, Ghalia Musalam Salim; Dr. Omar, Khaled Abdalgader M., and Ms. Alsinani, Maryam Juma. (2017).*A Rule-Based System for Advising Undergraduate Students*

Nambiar, Arun N., Dutta, Anish K. (2010). *Expert System for Student Advising using Jess*

Olawande Daramola, Onyeka Emebo, Ibukun Afolabi, Charles Ayo (2014). *Implementation of an Intelligent Course Advisory Expert System*

Pandey, Shubham;Sharma, Vivek and Agrawal,Garima. (2019). *Modification of KNN Algorithm*

Petkovic, Matej; Boumghar ,Redouane; Breskvar, Martin; Dzeroski, Saso; Kocev, Dragi; Levatic,Jurica; Lucas, Luke; Osojnik; Aljaz; Zenko, Bernard and Simidjievski1,Nikola. (2019). *Machine learning for predicting thermal power consumption of the Mars Express Spacecraft*

Pintelas , Panagiotis and Livieris , Ioannis E. (2020). *Ensemble Algorithms and Their Applications*

Raed Shatnawi, Qutaibah Althebyan, Baraq Ghalib, Mohammed Al-Maolegi (2015). *Building a Smart Academic Advising System Using Association Rule Mining*

Rajaguru, Harikumar and Chakravarthy, Sannasi S R. (2019). *Analysis of Decision Tree and K-Nearest Neighbor Algorithm in the Classification of Breast Cancer*

Rampisela, T V and Rustam,Z. (2018). *Classification of Schizophrenia Data Using Support Vector Machine (SVM)*

Reddy, M. Sravan Kumar; Kumar, K.E. Naresh; Singh Rajput, Dharmendra. (2019). *Improving ADA-boost as a Popular Ensemble in Classification Problem*

Samay Abu Naser; Ihab Zaqout; Mahmoud Abu Ghosh; Rasha Atallah; Eman Alajrami. (2015). *Predicting Student Performance Using Artificial Neural Network*

Sethi, Kapil; Jaiswal,Varun; Ansari and Mohd Dilshad. (2018). *Machine Learning Based Support System for Students to Select Stream (Subject)*

Shin, Hyungsik and Park, Jeong Yeop. (2018). *Automatic Task Classification via Support Vector Machine and Crowdsourcing*

Shubham, Jinde. (2018). *Ensemble Learning — Bagging and Boosting*

Styawati ,Styawati and Mustofa,Khabib. (2019). *A Support VectorMachine-Firefly Algorithm for Movie Opinion Data Classification*

Tirumala, Sreenivas Sremath;Sarrafzadeh, Abdolhossein and Ali,Shahid. (2015). *Ensemble Learning Methods For Decision Making: Status and Future Prospects*

Upah, Sylvester Charles. (2016). *The Impact of predictive analytics based advising on the Selection and Change of Major among first year, first-term students in engineering.*

Victoriano, Jayson M.; Lacatan, Luisito L., and Vinluan, Albert A.. (2020). *Predicting River Pollution Using Random Forest Decision Tree with GIS Model: A Case Study of MMORS, Philippines*

Villaverde ,Mónica , Aledo, David, and Perez, David (2019). *On the Adaptability of Ensemble Methods for Distributed Classification Systems: A Comparative Analysis*

Walid Mohamed Aly; Osama Fathy Hehazy; Heba Mohmmed Nagy Rashad. (2013) *Automated Student Advisory using Machine Learning*

Zhang, Li. (2018). *Early Identification of At-Risk Students Using Iterative Logistic Regression*

Zhang, Shichao; Li, Xuelong; Zong, Ming; Zhu, Xiaofeng and Wang, Ruili. (2017). *Efficient kNN Classification with Different Numbers of Nearest Neighbors*

Zhang, Zhongheng. (2016). *Introduction to Machine Learning: k-Nearest Neighbors*

Zhiyuan, Ma, Wang, Ping, Gao, Zehui, Wang, Ruobing, and Khalighi , Koroush, (2018). *Ensemble of Machine Learning Algorithms using the Stacked Generalization Approach to Estimate the Warfarin Dose*