Presented to the College of Computer Studies

De La Salle University - Manila

Term 2, A.Y. 2022-2023

In partial fulfillment of the course

In CSINTSY S14

Major Course Output 3: Machine Learning

Submitted by:

Balderosa, Ernest

Caasi, Samantha Nicole

Marcellana, John Patrick

Noche, Zach Matthew

Submitted to:

Thomas James Tiam-Lee, PhD

April 17, 2023

I. Introduction

The Major Course Output 3: Machine Learning of Introduction to Intelligent Systems (CSINTSY) course requires implementing Machine Learning analysis by applying two machine learning algorithms for a classification task given a specific data set.

The problem statement in this course output revolves around how a student is recognized as being employable or less employable. Employable is a term that describes the ability of a person to work in a job or industry (Dawson, 2022). A person's employability refers to their skills, behavior, and knowledge and how they are seen on a person by the employer. Some key factors will make a person employable, such as the ability of a person to learn and adapt rapidly, their capacity for effective communication and interpersonal interaction, their work ethic, their attitude, and their willingness to work with others (Kelley, 2023). While the qualities to be considered less employable are the opposite such as having poor academic performance, lacking experience, and poor communication skills (Kelley, 2023).

In today's market, when employment is often changing, and new industries are developing, employability is becoming increasingly crucial. A degree or a certain set of technical skills is no longer sufficient. Candidates with a variety of soft talents and character traits that would help them to thrive in a fast-paced, dynamic workplace are in demand by employers.

According to the study by Hosain (2021), the factors: of academic performance, communication skills, personality, and teamwork and problem-solving skills greatly affect the employability of a graduate student in Bangladesh. This study may be a basis for the specific problem statement.

Academic performance is a term used to describe how well a student is performing in their academic work. It is frequently used to gauge a student's proficiency in a particular subject or ability and is typically quantified by grades or test results. Inborn talent, drive, effort, and outside variables like family background and the socioeconomic situation can all impact academic performance. Since factors beyond a student's control may affect their capacity to succeed in their studies, it is crucial to remember that academic success is not necessarily an accurate indication of a student's potential (Tadese, 2022).

Communication Skills are the skills needed to effectively and efficiently deliver information. The process of transferring ideas, thoughts, or information between people, groups, or organizations is known as communication. Various skills, including verbal and nonverbal communication, active listening, questioning, and feedback, are necessary for effective communication. These abilities are crucial for establishing connections, increasing comprehension, and attaining shared objectives (Rubble, 2018).

Personality is a person's distinct combination of features, qualities, and actions. It covers traits like attitudes, values, hobbies, and emotional dispositions. A person's ideas, feelings, and behaviors can be affected by their personality traits, which can be innate and learned. There are several ways that personality can impact employment. The personality qualities conscientiousness, emotional stability, agreeableness, and openness to new experiences are among those that many businesses look for in job candidates (University College London, 2021).

Since many positions demand people to collaborate with coworkers to accomplish a similar goal, teamwork skills are highly valued in the workplace. Companies seek candidates that can work well in a team setting, communicate well, and positively contribute to the team's success. Those with strong cooperation abilities can foster excellent working connections with their coworkers, boost productivity, and improve the workplace atmosphere. Employers place high importance on problem-solving abilities, too. Workers who recognize issues, assess the situation, and develop workable solutions are highly valued. An individual's capacity for independent thought, initiative, and problem-solving is demonstrated by their problem-solving ability. Increased productivity, better customer satisfaction, and a happier workplace are all possible outcomes of these talents (Fajaryati et al., 2020).

The motivation for this problem statement is its relevance to the group's situation. Soon enough, the group will face a similar situation in asking what qualities and skills to be recognized as employable. Especially on questions of what are the things to consider, and what are the things to improve on for employers to recognize students as being employable. Solving this question will be of great help not only to the group but to all students who are starting their job careers.

The problem statement aims to recognize a student's employability based on several independent features, such as their appearance, manner of speaking, physical condition, mental alertness, self-confidence, ability to present ideas, communication skills, and student performance rating. Therefore, the approach to solving this problem is to use a machine learning algorithm that approaches the problem as a binary classification problem. According to Wolff (2020), classification algorithms use input training data to predict the likelihood of subsequent data falling into one of the predefined categories. Therefore, this problem statement is well-suited for a classification problem because employability can be derived as the label or the predefined categories from the given dataset, meaning it is a target variable that the machine learning model will learn from to predict whether a student, based on their independent features, is employable or not.

Furthermore, classification algorithms are often used for solving problems where the output variable is categorical. A classification algorithm aims to find a relationship between the input variables (independent features) and the output variable (employability). The algorithm uses historical data to learn patterns and relationships in the data and then applies that learning to make predictions on new data. In this case, a classification algorithm can be used on a machine to be trained on the given dataset and make predictions on new data to determine if a student is employable based on their independent features. By doing so, the group can identify factors contributing to employability and help students improve their employment chances.

The possible benefits of solving this problem include a better understanding of the factors contributing to employability. This may improve the students' outcomes in finding employment and building successful careers. Additionally, employers will benefit from a more accurate and efficient method of determining the employability of job applicants through objective and data-driven judgment. This, in turn, can impact the economy of the Philippines positively.

II. The Dataset

The group chose the "Students' Employability Dataset - Philippines" dataset from Kaggle. The dataset comprises 2,982 mock job interview results collected across various university agencies in the Philippines (Hamoutni, 2021). The dataset was obtained from 2015 to 2018. The dataset has eight independent features: General Appearance, Manner of Speaking, Physical Condition, Mental Alertness, Self-Confidence, Ability to Present Ideas, Communication Skills, and Student Performance Rating. The classes are: 'Employable' and 'Less Employable', respectively, representing whether the student is considered employable or less employable based on independent features. The source of the dataset was through the following link:
https://www.kaggle.com/datasets/anashamoutni/students-employability-dataset

The given independent features of the dataset uses a numeric metric system to describe the attributes of a student to determine their label class. The independent features or variables are called ordinal data in a more formal definition. According to Bhandari (2022), ordinal data is classified within a variable to represent a natural rank in order, but the distance between the categories is unclear. In the context of the dataset, using the independent feature "General Appearance" and "Physical Condition," there is meaning in the difference between a student rated a value of 5 in "General Appearance" compared to a student who is rated a value of 3. However, this does not mean it uses the same scale or distance for the independent feature "Physical Condition." Lastly, how the independent features were rated for each student in the dataset was not explained on the Kaggle website; thus, the interpretation of the ratings per independent feature will be treated as is.

The label class in the dataset represents the target variable, which is the dependent feature the model will try to predict using the independent features. The relation of the target variable to the independent feature depends on how the student was rated on each feature. In inference, highly rated students are more likely to be labeled "Employable," while those generally rated below average are more likely to be labeled "Less Employable." The data set using a binary class label is important in the context of the goal of the group, as this helps frame the problem as a binary classification task for which a wide range of machine learning algorithms can be used. It can also help interpret the result easily as it only has two possible outcomes to analyze and evaluate the performance of the model to be used.

It is important to understand the dataset to get a complete picture of its distribution. One such way is to calculate the mean and standard deviation of the dataset. The mean and standard deviation are important terms in statistics used to measure the approximate center of the data and how close or dispersed the spread of the data to the mean, respectively. According to Zach (2022), understanding the mean is essential because it represents the average value of the data set, which will help in the further observation of data as the approximate "center" of the dataset is located. Standard deviation is then used to measure the spread of values in a sample, where higher values represent higher spreads. In comparison, lower values represent the data as tightly packed or close to each other. Knowing the center's location and the values' spread will help better understand the distribution of values in any dataset (Zach, 2021).

Given the dataset, below is an enumeration of the mean and standard deviation of each independent feature's numerical values:

Table 1. Mean and Standard Deviation of the Dataset

INDEPENDENT FEATURE	MEAN	STD DEVIATION
GENERAL APPEARANCE	4.246814	0.678501
MANNER OF SPEAKING	3.884641	0.757013
PHYSICAL CONDITION	3.972166	0.744135
MENTAL ALERTNESS	3.962777	0.781982
SELF-CONFIDENCE	3.910798	0.807602
ABILITY TO PRESENT	3.813883	0.73939
COMMUNICATION SKILLS	3.525486	0.743881
STUDENT PERFORMANCE RATING	4.610664	0.692845

The center of each independent feature is between the range of 3.53 to 4.61; this indicates that the central tendency of each feature is relatively similar. Conversely, the standard deviation of each feature is between the range of 0.68 to 0.81, which also indicates that the spread or dispersion of the data of each feature is relatively similar. Given that the numeric range of each feature is from 1 to 5, a standard deviation in the given range suggests that the range is somewhat spread out but not extremely so. Below is a plot showing each feature's mean and standard deviation for better understanding.

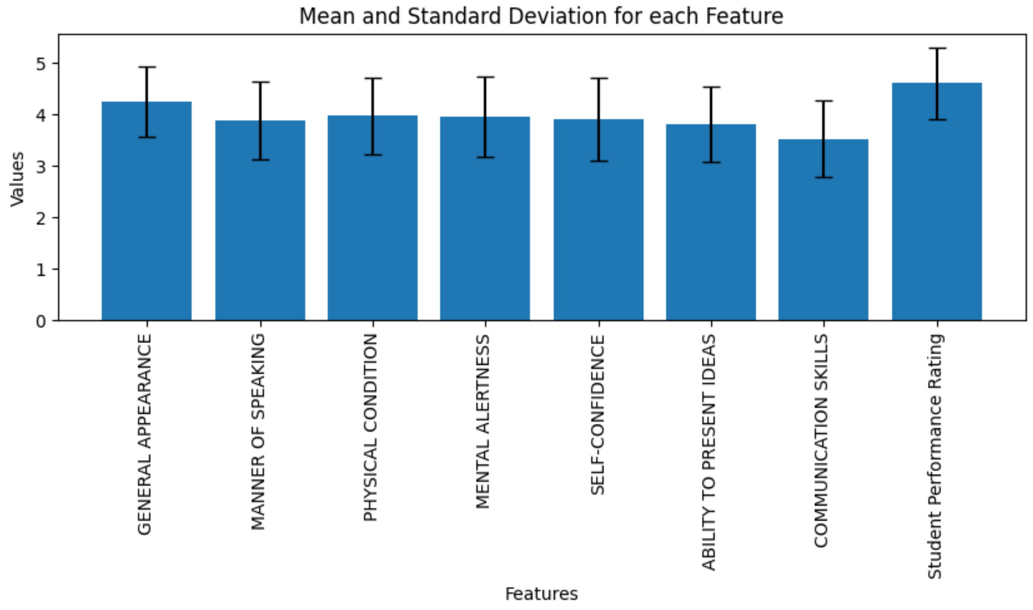


Figure 1. Bar plot of the mean and standard deviation of each feature

As visualized in Figure 1, all the features have similar central tendencies and standard deviation; this implies that each of the independent features is comparable, thus, may have implications for the performance of the machine learning algorithm to be used.

The distribution of the binary label class or target variable was also analyzed to see if there is an imbalance between employable students and those less employable; the figure below shows the distribution of the label class.

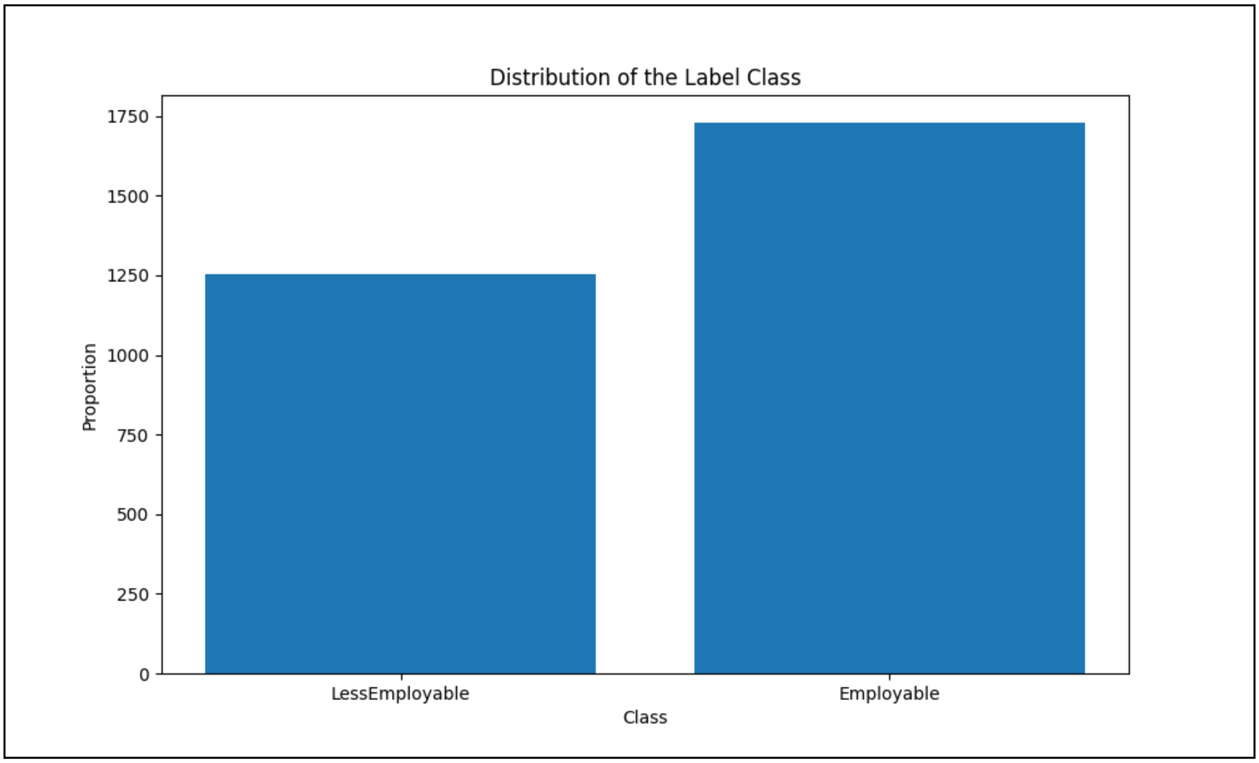


Figure 2. Bar plot of the distribution of the binary label class

Examining Figure 2, it can be seen that there are a bit more students labeled as employable as compared to being less employable. The exact numbers of the label for employable and less employable are 1729 and 1253, or 57.98% and 42.02%, respectively. This implies that there is a slight imbalance between the label class. Thus, preprocessing techniques might be needed to handle the imbalanced distribution to make it easier for the algorithm to learn from the data.

Nevertheless, the dataset is sound and should be easy to learn for any classification algorithm. The purpose of getting the mean, standard deviation, and distribution were to determine if there is a need to preprocess the data and what specific preprocess techniques must be used. Based on the results, there is no urgent cause to use popular preprocessing techniques such as normalization and standardization. Although it would still improve the learning models' performance, it would not give significant performance gains. The same could be said for the imbalanced dataset, there might be a need for resampling techniques, but further analysis will be needed.

III. Methodology

The group utilized a machine learning framework called sci-kit learn. A machine learning framework refers to software packages or tools that offer a selection of pre-built features and interfaces for creating and deploying machine learning models (Rowe & Johnson, 2020). It is simpler for developers to create and train models for various applications because they provide an abstraction layer over the low-level specifics of machine learning techniques (Rowe & Johnson, 2020).

According to Rouse (2019), sci-kit-learn is a well-known open-source Python toolkit for machine learning tasks, including classification, regression, and clustering. It is based on NumPy, SciPy, and matplotlib and offers a variety of effective tools for data mining and data analysis. Scikit-learn is well-known for its user-friendly interface and comprehensive documentation and is used extensively in academia and business (Rouse, 2019). Scikit-learn was chosen as a machine learning framework for this output because it already provides two algorithms that will be used: Logistic Regression and Decision Tree (Rouse, 2019).

The implementation of learning models in sci-kit-learn, in general, has two basic, abstracted implementations: those are the training function (a fit function that will fit the training data to the specific model) and the prediction function, which will take in new data (in this case, the test subset that was split beforehand) and then apply the model's newly learned parameters from the training by applying the mathematical equation specific for a model and then outputting the predicted labels (either 1 or 0). Although the group is not knowledgeable about how each model actually fits data and uses it to tune the mathematical equation it uses to predict labels, it is still important to distinguish between how Logistic regression and Decision Tree models work at their surface level. This will be discussed further in later sections.

The group utilized a machine-learning pipeline with the steps described in the flowchart below. This will help visualize the overall steps in describing the machine learning process.

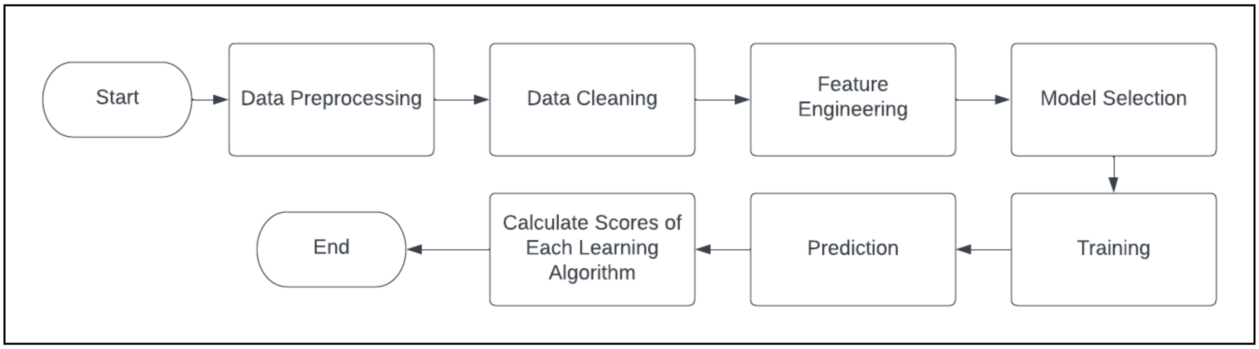


Figure 3. Machine Learning Pipeline

The dataset used first undergoes data preprocessing steps in which label encoding was specifically used. It is a method for transforming categorical data into numerical data that can be used in machine learning models. Each category in a categorical variable must be given a special numerical label (Yadav, 2019)

Other preprocessing steps like scaling and resampling techniques were considered because of the imbalances in the data as explained where there are a bit more students labeled as employable as compared to being less employable. The group initially performed these data pre-processing steps. However, the improvement was negligible and, at best, marginal.

Feature engineering is applied in the data set as well. In feature engineering, the relevant characteristics from the raw data that can be utilized as input in a machine-learning model are chosen and transformed (Patel, 2021). By developing a collection of features, the machine learning model will perform better and will better capture the pertinent information in the data (Patel, 2021).

The group utilized two machine learning algorithms, namely, Logistic Regression and Decision Tree. The goal of Logistic Regression, a statistical method used for binary classification tasks, is to predict the probability of an event occurring (IBM, n.d.). It is a type of regression analysis that models the relationship between a target variable and one or more independent features. The target variable usually takes the form of binary outcomes such as “0” or “1”. On the other hand, independent features can be continuous, discrete, or a combination of both. The Logistic Regression model applies the sigmoid function to the linear combination of the independent features and their coefficients; this transforms the output into a probability value between 0 and 1. The probability value is then used to make predictions about the outcome variable. A threshold value of 0.5 is usually applied where probabilities greater than or equal to 0.5 are classified as one category, and probabilities less than 0.5 are classified as the other category.

Applying Logistic regression to the problem at hand, the model will start by calculating the relationship between each independent feature and the target variable. The independent features, as aforementioned in the dataset portion of the paper, are used to model the relationship with the target variable, employability. The output of the Logistic Regression model will then be a predicted probability of the student being ‘Employable’ or ‘Less Employable’ based on their performance in job interviews. Furthermore, a threshold value of 0.5 can be used to classify the student into one of the two categories.

On the other hand, one common supervised learning technique used in machine learning to create a prediction model is the Decision Tree. It is a visual representation of every option for a choice based on specific circumstances (Gupta, 2017). It shows a number of choices and their potential outcomes using a tree-like paradigm (Gupta, 2017). The Decision Tree algorithm divides the input data recursively into subgroups according to the parameters of the input data. A decision made on the basis that corresponds to one of the input attributes is represented by each internal node in the tree, as well as the anticipated output is represented by each leaf node (Gupta, 2017).

The algorithm works by placing the complete data set just at the root node in the tree when the algorithm starts. Each time, the algorithm chooses a feature based on a certain parameter, including information gain or Gini impurity, that effectively separates the data (Chauhan, 2022). The procedure is repeated for every subset until a halting requirement, such as a maximum tree depth or a minimum sample size per leaf node, is satisfied. The data is then divided into subsets according to the value of that feature. Based on the majority class or mean value of the training samples that reach that node, the algorithm provides a forecast for the target variable at each leaf node (Chauhan

2022). The resulting tree has a hierarchical structure with leaf nodes representing predictions for the target variable and inside nodes representing decisions based on features (Roy, 2020). Using the values of the features in the new data, one can traverse the tree from the root node down to a leaf node and return the forecast at that leaf node to create predictions for new data.

These algorithms were chosen by the group since the problem at hand is a binary classification problem. Having that said, Logistic Regression and Decision Tree are both appropriate choices as they can both predict the probability of a binary outcome in which students will be classified as either “Employable” or “Less Employable”. Furthermore, as the group has yet to delve into the inner workings of machine learning models, they have opted to work with models that are more interpretable. Having a highly interpretable model gives us a way to immediately understand the model’s performance and how it makes predictions on the data.

The group saw that Logistic Regression and Decision Trees are reasonable for this output since for binary classification problems, where the objective is to estimate the likelihood that an observation belongs to one of two potential classes based on a set of predictor factors, both algorithms are widely used statistical techniques.

A probability value with a range of 0 to 1 that represents the likelihood that an observation belongs to the positive class is the output of a Logistic Regression model (Raj, 2021). A linear model that is simple for laypeople to comprehend and analyze is Logistic Regression. The influence of each predictor variable on the likelihood of a successful outcome can be calculated using the model's coefficients. It is a versatile tool for a variety of data sets because it can handle a mix of categorical and continuous predictor variables (Raj, 2021). A probability score between 0 and 1 is generated by logistic regression, and this score can be used to make decisions and determine how definite the predictions are.

In addition, due to the ability of Decision Trees to handle both category and quantitative information, Decision Trees are a common approach for classification tasks (Yang, 2019). Decision Trees can be used to pinpoint the most crucial elements that influence a student's likelihood of finding employment in the context of graduate students' employability. Decision Trees are also generally simple to interpret, which makes it simpler for recruiting managers or employers to comprehend the decision-making process (Yang, 2019).

IV. Results and Analysis

In machine learning, it is important to use appropriate metrics to evaluate the performance of a model. There are many different metrics available for evaluating different types of models. And it is important to choose the right metrics to avoid poor performance when the model is applied to new data. For example, in a binary classification problem where there are two possible output classes (such as “employable” and “less employable”), it is common to represent the output as either 1 (positive) or 0 (negative). To evaluate the performance of a binary classification model, it is important to understand the concepts of true positives (TP), false negatives (FN), true negatives (TN), and false positives (FP). These terms refer to the following:

- True Positive (TP): Samples that are correctly predicted as positive
- False Negative (FN): Samples are predicted as negative but are actually positive
- True Negative (TN): Samples that are correctly predicted as negative
- False Positive (FP): Samples are predicted as positive but are actually negative

By considering these four outcomes, the group can properly asses and calculate the performance of the chosen binary classification models. Thus, the metric to be used are metrics that are commonly used in binary classification problems. These are Accuracy, Precision, Recall, F1-Score, and Confusion Matrix. Each metric will be expounded further.

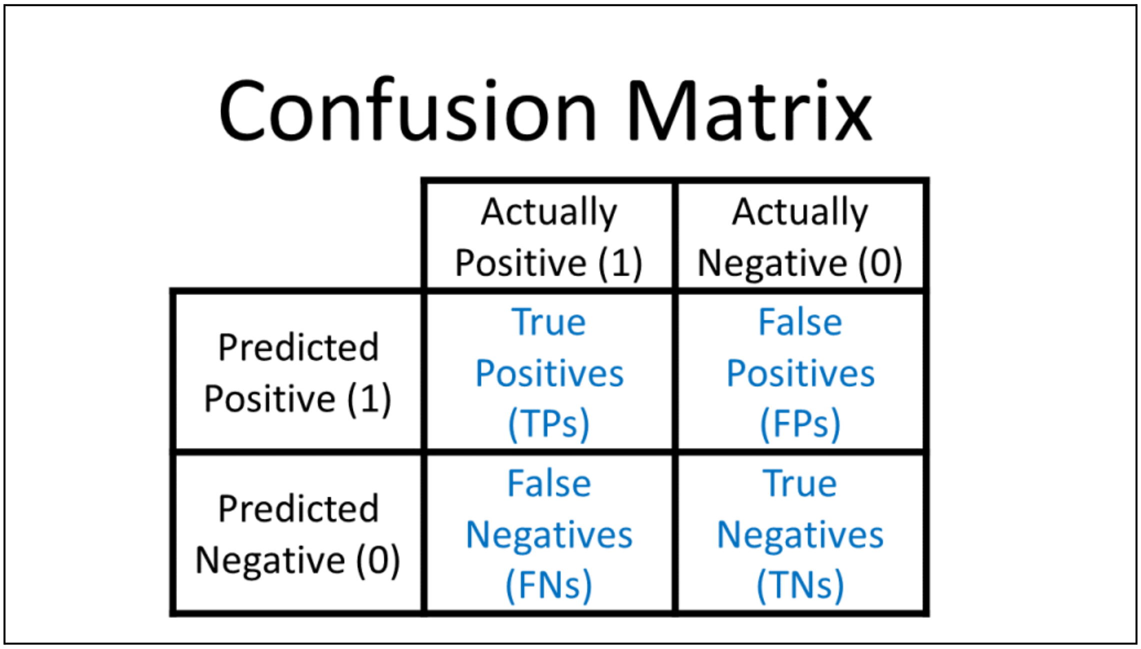
According to Agrawal (2022), a model’s accuracy is simply how well it is able to predict the sample’s classification correctly. To get the ratio or score of accuracy, simply sum up all correct prediction outcomes (TP and TN) and then divide it by the entire prediction made by the classifier model. Accuracy is commonly a metric used to get a general baseline performance of the model as it is easy to interpret. However, using accuracy only as a metric does not guarantee the full picture of the model’s performance. In a situation where the dataset has a huge imbalance in the class (there are more positives than negatives and vice versa) the model could always be predicting a positive outcome and it would get an accuracy of 90% or higher. At face value, this can be seen as a good performance, but in reality, the test sample it was predicting contained more positive outcomes rather than negatives. If it were to encounter a balanced test sample, its accuracy score would plummet. Thus, it is important to consider other metrics, such as precision.

According to Agarwal (2019), precision simply means “How many predicted positive outcomes, are actually positive?” Thus, to get the score or ratio of precision of a model, simply get the number of true positive predictions (TP) and then divide it by the sum of all positive predictions (TP and FP). Precision as a metric is desirable when it is important for the model to correctly predict something as positive, and false positive outcomes are seen as undesirable. High precision, however, does not mean that the model is able to capture the actual positive outcomes of a given sample. It simply means that all predicted positive outcomes are actually positive but does not consider all the other outcomes it predicted as negative but are actually positive. In this case, the recall metric is used to evaluate the performance of a model predicting all positive cases.

According to Google Developers (2022), recall simply asks the question, “What proportion of actual positive was correctly identified?” Recall is a metric that evaluates a model if it can capture all positive outcomes of a given sample set. To get the recall score or proportion, simply get the proportion of true positive predictions (TP) and then divide it by the sum of true positive and false negative predictions (TP and FN) If, in precision, false positives are undesirable, in recall, false negatives are undesirable. This often means recall and precision are in a sort of tug-of-war state, meaning, aiming for high precision often lowers recall score and vice versa (Google Developers, 2022). This is because when a model aims for high precision, it wants to make sure that its positive prediction outcome is actually positive; this means the model tends to be conservative. On the other hand, aiming for high recall means the model will aggressively label positive outcomes, the model will be able to capture more actual positive outcomes but at the cost of precision. Thus to solve this back and forth, the F1-score metric is used to find the harmonic balance between precision and recall.

According to Korstanje (2021), F1 seeks to find harmony between precision and recall. It wants to get the harmonic mean or average between the score of precision and recall. Thus, the more equal and high the precision and recall scores are, the higher the score for the F1-score metric. To calculate the F1-score, the formula is given as $2\,\frac{Precision * Recall}{Precision + Recall}$. F1-score is also able to handle imbalanced datasets well because unlike accuracy that cannot distinguish whether it is predicting the majority or not, F1-score is a combination of precision and recall where it now says the question “ How many positive predicted outcomes are actually positive and how many actual positives were correctly identified?” Thus it is a useful metric to use for any kind of dataset.

Given all the metrics and their definitions, a useful tool to easily evaluate each metric score is by using a confusion matrix. A confusion matrix is a performance measurement table for problems where the output can be two or more classes (Narkhede, 2019). Recalling the 4 possible prediction outcomes of a model, the confusion matrix makes it easier to interpret all predicted and actual outcomes of a given sample set, which makes it easier to evaluate the 4 metrics mentioned previously. Below is a visualization of a confusion matrix:



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 4. Confusion matrix. Adapted from “Measuring Performance: The Confusion Matrix,” by Glass Box Medicine, 2019.

Once knowing the metrics to be used, it is necessary to use these metrics to evaluate the performance of Logistic Regression and Decision Tree models. Python and

Sci-kit were used as the primary programming language and library to implement the machine learning pipeline. Plotting libraries were also used to visualize the performance metrics of both models. In the code, the first step was to preprocess the target variable by using label encoding such that the label class “Employable” and “Less Employable” is encoded into 1 and 0, respectively. This is to say that if the model were to predict a positive outcome, it means that its prediction for a student is “Employable,” or else its “Less Employable.” the next step was to feature engineer the dataset. Correctly splitting the independent features and storing them in a variable “X” and also storing the target variable in variable “y.” After which is to split both the independent features and the target variable to test and train the data subset given a split ratio. Finally, the test subset will be used to train the Logistic Regression and Decision Tree models.

Before the actual training of the models start, it was decided in the group to have simple hyperparameters to control. Starting at the split ratio for training and testing data subset, there are multitudes of studies that suggest optimal split ratio, from an 80:20 ratio, 70:30 ratio, or even a 50:50 ratio (Joseph, 2022). Although there are mathematical ways to optimally find the ratio for a given data set, the group decided to simplify the hyperparameter used to control the ratio of training and test data split by using a split ratio of 75:25, 75% training, and 25% test data split.

For Logistic Regression, there exists a multitude of hyperparameters for data scientists to control, but for the sake of simplicity, only the hyperparameter that controls the iteration of the training will be changed. The hyperparameter “max_iter” simply means how often the model will update the coefficients or weights during the minimization process before the solvers converge. Setting this hyperparameter too low will prevent convergence from happening, which will lead the model to do suboptimal solutions while setting it too high will make the training process take too long to finish without any significant performance gains.

For the Decision Tree, it is the same as Logistic Regression, where it has an abundance of hyperparameters to control, but the group decided to only change the hyperparameter for the Decision Tree’s search depth, in other words, the max depth that determines how deep the Decision Tree is starting from the root node, up to the leaf nodes. Setting the max depth hyperparameter too low will result in the Decision Tree being unable to capture all relationships of each independent feature and, thus, create an under-fitted tree with suboptimal performance. On the other hand, setting the max depth too high will result in overfitting, which means the model fitted the sample training data too well and will perform worse on new, unforeseen data as its decision rules are strict.

For the following analysis of plots, the hyperparameter for the Logistic Regression model was set to do a max of 1000 iterations. This number was decided upon as iterations above this threshold yielded little to no performance gains. In the Decision Tree, however, the group had to do extra analysis to decide the best hyperparameter value for max_depth so that it does not underfit or overfit the sample training data. The group decided on a hyperparameter value for a max depth of 10 for now, as it is correlated with the number of independent features the data set has, however, further analysis is needed to determine if the decided max depth is optimal or not.

After training and getting the prediction score, below is a graph to show the score for each metric of Logistic Regression and Decision Tree model in a side-by-side comparison fashion.

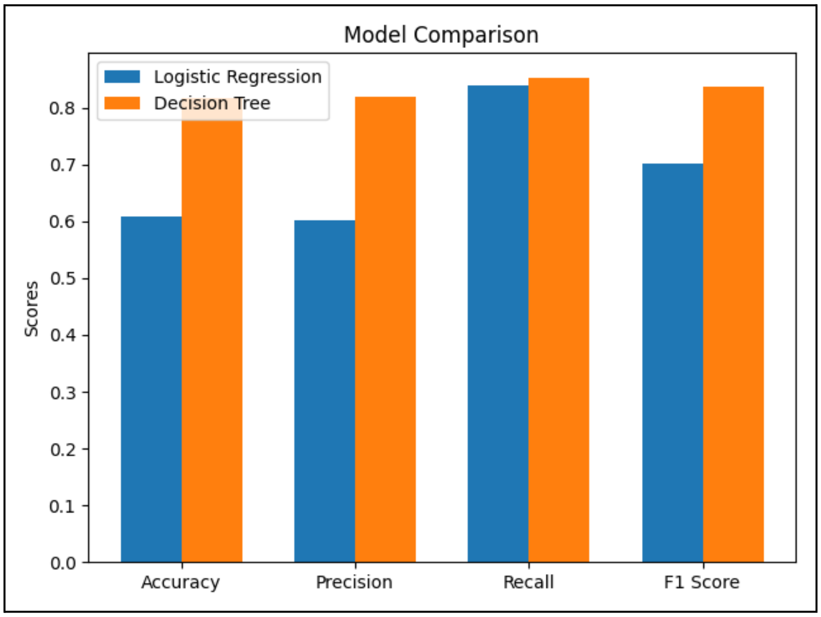


Figure 5. Logistic Regression and Decision Tree Model Metric Comparison

According to Zach (2022), there is no definite answer as to what an acceptable score for metrics such as accuracy, but 50% is seen to be the bottom baseline of performance as this means the model is predicting the labels half of the time. In the group's case, the group decided these score ranges to be the interpretation of scores:

- 50-60% "Functional." The model works but is undesirable.
- 60-70% "Ok" The model's performance is acceptable but could be better
- 70-80% "Good" The model's performance is good enough
- 80+% "Great" The model's performance is great and is expected

In Figure 5, Logistic Regression seems to be performing worse as compared to Decision Tree where it has "Great" performance in all its metrics, while Logistic Regression has "Ok" performance on accuracy and precision, "Great" for recall, and "Good" for F1-score. For a more defined score for each metric score of the models, below is the table tabulating the score for each metric.

Table 2. Scores for Each Metric of The Models

METRIC	**LOGISTIC REGRESSION**	**Decision Tree**
ACCURACY	0.608579088	0.816353887
PRECISION	0.602811951	0.819248826291079
RECALL	0.838630807	0.853300733
F1-SCORE	0.701431492842535	0.835928144

As seen in the tabulation, the bar graph visualization is true to the actual score calculated for each metric of the models. Thus, there is a need first to analyze the confusion matrix for Logistic Regression to understand further the scores it has for each metric.

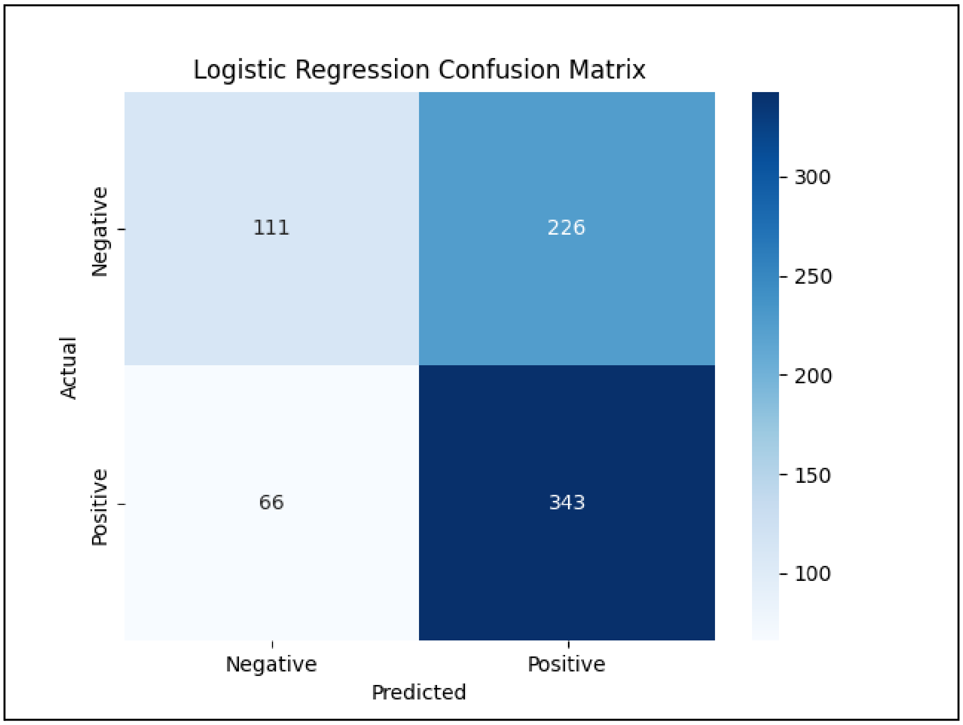


Figure 6. Confusion Matrix of Logistic Regression

As Seen in Figure 6, the confusion matrix for the Logistic Regression model is visualized to show samples that the model predicted as either negative or positive and those that are actually negative or positive labels. For easier interpretation, the prediction count for each possible outcome is as follows:

- True Positive (TP) - 343
- True Negative (TN) - 111
- False Positive (FP) - 226
- False Negative (FN) - 66
- Total Sample Size - 746

Analyzing the accuracy of the Logistic Regression, accuracy score is calculated as $\frac{343 + 111}{746} = 0.608$. This score is exactly the score seen in Table 2. The accuracy metric signifies that the model is 60% correct at its prediction most of the time; this again means that it has "Ok" performance but could be better. Given the proportion of true positives predictions to true negatives is higher, that means the model seems to be aggressive enough to label samples as positive rather than negative. This is indicative to its precision and recall score. Signifying that it has higher recall than precision.

Calculating both precision and recall, the precision score is calculated as $\frac{343}{343 + 111} = 0.602$ while recall is calculated as $\frac{343}{343 + 66} = 0.838$, both scores are as seen in Table 2. Now it can be definitely be said that the model is labeling samples as positive (“Employable”) more compared to labeling them as negative (“Less Employed”). While the model’s recall is “Great”, its precision is “Ok”.

To better understand what this means, calculating the F1-score will help interpret the model’s performance as a whole. The F1-score is calculated as $2\frac{0.602 * 0.838}{0.602 + 0.838} = 0.70$, a score approximate to the score seen in Table 2. In this range, it is interpreted as “Good”, this means that despite having a lower-than-expected score for precision, because the model has a high score for recall, the average score then says that overall, the model’s performance is “Good” and acceptable to be used to predict a student’s employability.

Nonetheless, the group decided to analyze a bit further as to why the Logistic Regression model is performing worse compared to the Decision Tree model. There can be many reasons as to why, it could be because of the imbalance of the dataset as seen in Figure 2, where there are more positive labels compared to negative, which might have led the model to predict the majority more. It could also be that the model cannot exactly capture the relationship between the independent features and the target variable may be due to it being a non-linear relationship.

The group then decided to first check how the model trained itself to the training set, by plotting out the coefficients it calculated per independent feature to know the magnitude of the odds that affect its prediction for positive or negative labels and to know what independent features have the biggest influence on the outcome.

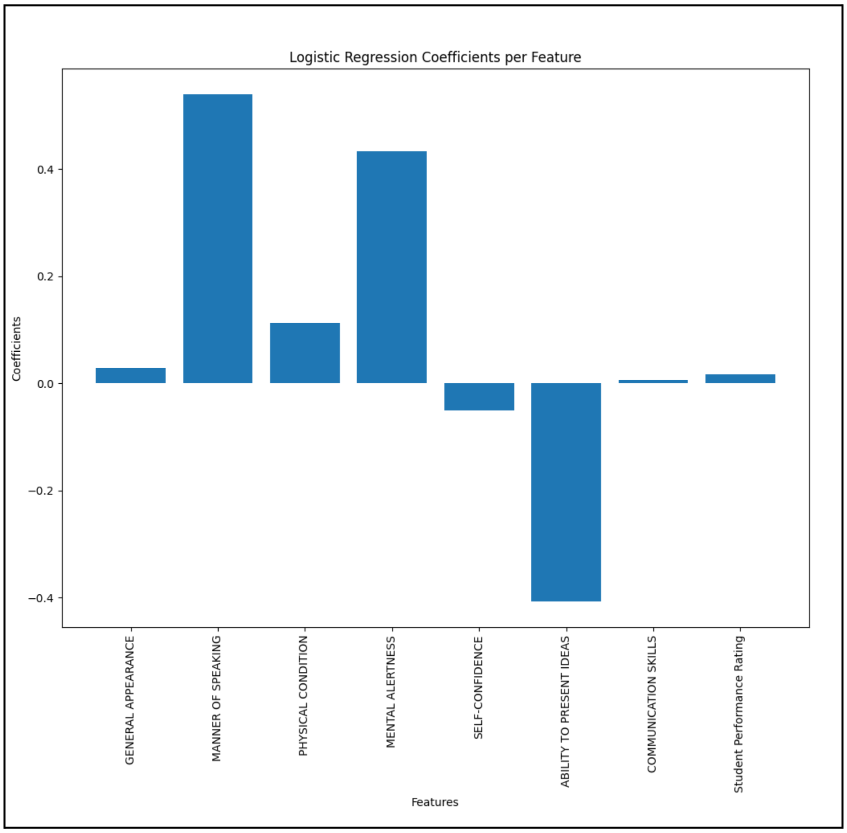


Figure 7. Bar Graph of The Coefficient Calculated by the Logistic Regression Model per Feature

As seen in Figure 7, the coefficients were calculated during the model's training phase. To formally define what coefficients are, coefficients in Logistic Regression are a representation of the magnitude and direction of the influence that each independent feature has on the target variable. Specifically, they indicate a logarithmic change in the odds of the outcome (Kisselev, 2021).

Simply put, the coefficient of each independent feature can be interpreted as the logarithmic odds of an outcome. In the figure above, the plot shows the direction and magnitude of the coefficient per feature, indicating the importance of each feature in predicting the target variable label. The higher the magnitude of the coefficient, the stronger the influence of that feature on the outcome, and the direction of the coefficient indicates whether it would predict a positive or negative outcome.

Analyzing the bar graph shows that 3 out of the 8 independent features are seen as important in predicting the outcome of the target variable. Specifically, the independent feature “Manner of Speaking” and “Mental Awareness” are calculated by the model as important in predicting the odds of a sample to be a positive outcome. On the other hand, the independent feature “Ability to Present Ideas” is calculated by the model as important in predicting the odds of a sample to be a negative outcome. Thus, it can be said that the relationship between the independent feature and the target variable may be non-linear since Logistic Regression was not able to capture all relationships, especially when only 3 out of 8 independent features have a significant influence on the target variable label. However, this is not to say that the actual relationship between the independent feature and the target variable is non-linear, but this is to signify that Logistic Regression has limitations in capturing the relationships present.

Although Logistic Regression works best when the relationship between the independent features and the target variable is linear, it can still capture non-linear relationships. This is because it uses the natural logarithm odds ratio rather than the probability itself. This means that the model can transform non-linear relationships into linear relationships. Nevertheless, there are still limitations to it, and the coefficients in Figure 7 show that it cannot capture all relationships present.

With this in mind, it is important to discuss the other model Logistic Regression was compared with, the Decision Tree model. Going back to Figure 5, it is to be iterated that the performance of the Decision Tree model is interpreted as “Great” across the chosen metrics. This means that the Decision Tree model was seemingly able to capture the relationship between the independent features and the target variable; thus, was able to predict the correct outcome in a very acceptable manner. To further see how well the Decision Tree performed, below is the confusion matrix of the model.

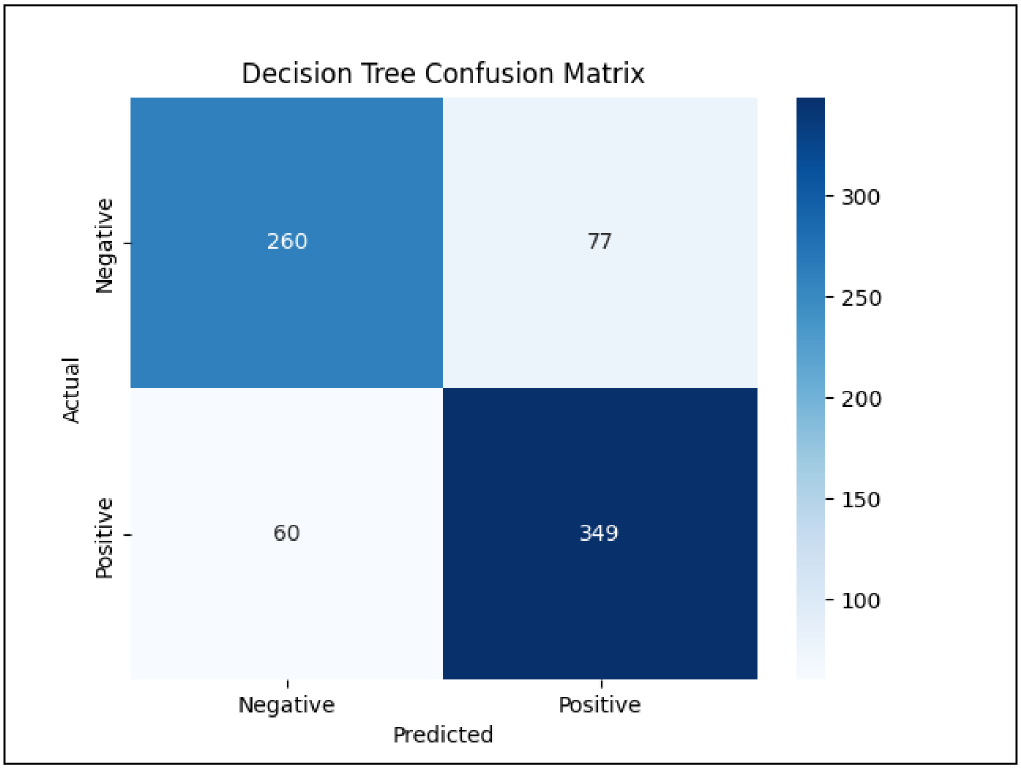


Figure 8. Decision Tree Confusion Matrix

Seen in Figure 8, the confusion matrix for the Decision Tree model is visualized to show samples that the model predicted as either negative or positive and those that are actually negative or positive labels. For easier interpretation, the prediction count for each possible outcome is as follows:

- True Positive (TP) - 349
- True Negative (TN) - 260
- False Positive (FP) - 77
- False Negative (FN) - 60
- Total Sample Size - 746

Analyzing the accuracy of the Decision Tree, accuracy score is calculated as $\frac{343 + 260}{746} = 0.816$. This score is exactly the score seen in Table 2. The accuracy metric for the Decision Tree is interpreted as "Great" because for all predictions made, at least 80% was correct and that is an impressive model. Nevertheless, accuracy cannot be the only basis of its performance and other metrics must be discussed. Still, it is a telling sign due to the fact that the ratio of true positive predictions and the true negative predictions are more evenly distributed as compared to Logistic Regression's prediction ratio. To further see the overall performance, precision, and recall are calculated.

Calculating both precision and recall, the precision score is calculated as $\frac{349}{349+77} = 0.819$ while recall is calculated as $\frac{349}{349+66} = 0.853$, both scores are as seen in Table 2. Both scores are interpreted as "Great" and are equal. This means that the Decision Tree model's positive predictions are correct and is also able to identify what samples actually have positive outcomes 80% of the time.

Given that both precision and recall are equal and have relatively high scores, the F1-score is then calculated as$2\frac{0.819 * 0.853}{0.819 + 0.853} = 0.835$, a score approximate to the score seen in Table 2. In this range, it is interpreted as "Great." Since both precision and recall have a high score and are relatively near each other, it is not surprising that the F1-score is also similar. This means that overall, the model is performing well given the dataset, despite its imbalance.

As done previously, the group decided to conduct a deeper analysis of the Decision Tree model's superior performance compared to Logistic Regression. One hypothesis for this performance could be that the Decision Tree is better at capturing the relationships between independent features and the target variable. Thus, to understand how the Decision Tree model learned from the dataset and captured the essential relationships to determine the rules for predicting the outcome label, the group planned to examine the model's abstracted internal workings. In addition, the group wanted to visualize the Decision Tree graph to understand better how the model predicts the outcome of a sample. Finally, the group aimed to investigate whether the chosen hyperparameters, specifically the maximum depth of the Decision Tree, might cause overfitting or underfitting on the training data. These further analyses could provide valuable insights into the Decision Tree's superior performance and could guide future model optimization efforts.

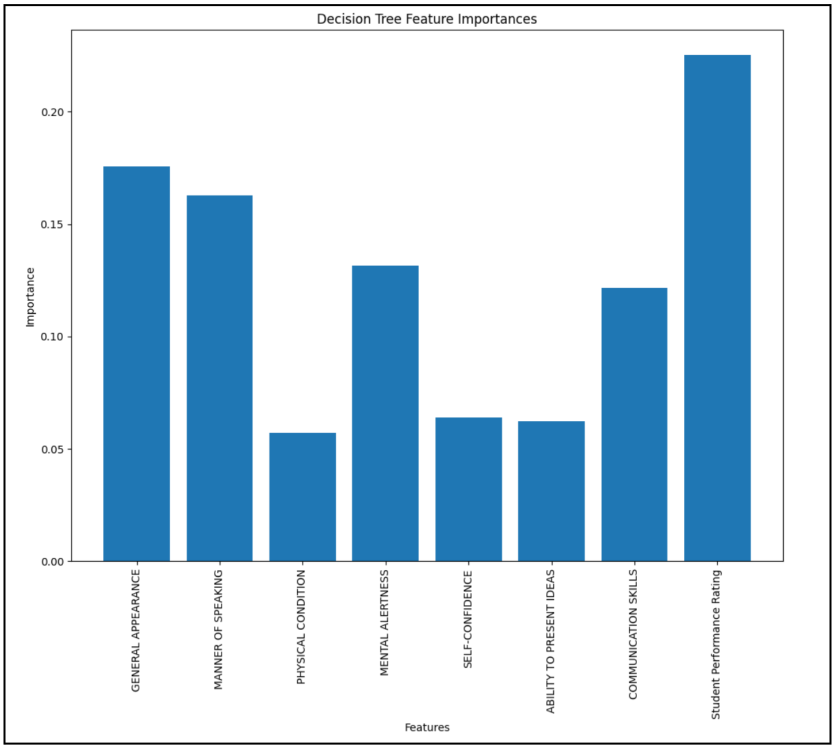


Figure 9. The Calculated Importance of the Decision Tree per Feature

Shown in Figure 9 is the calculated importance of each feature in a Decision Tree. The higher the importance score is, the stronger the influence of that feature on the outcome. Feature importance refers to the measurement of how relevant each feature or predictor variable was when the tree model was being built. It shows how much each attribute contributes to the choice that the tree ultimately makes (Bujokas, 2022). Feature importance are calculated through various methods including gini importance, permutation importance, and mean-decreasing impurity.

The graph in Figure 9 shows that the feature “Student Performance Rating” has the highest importance score which indicates it was the most significant feature for determining whether a student is employable or non-employable with the Decision Tree. In contrast, “Physical Condition”, “Self-Confidence”, and “Ability to Present Ideas” features were the least important using the Decision Tree as it has they have a close score with the lowest from the rest. This would indicate that these features had less impact on determining if a student is employable or less employable in a Decision Tree.

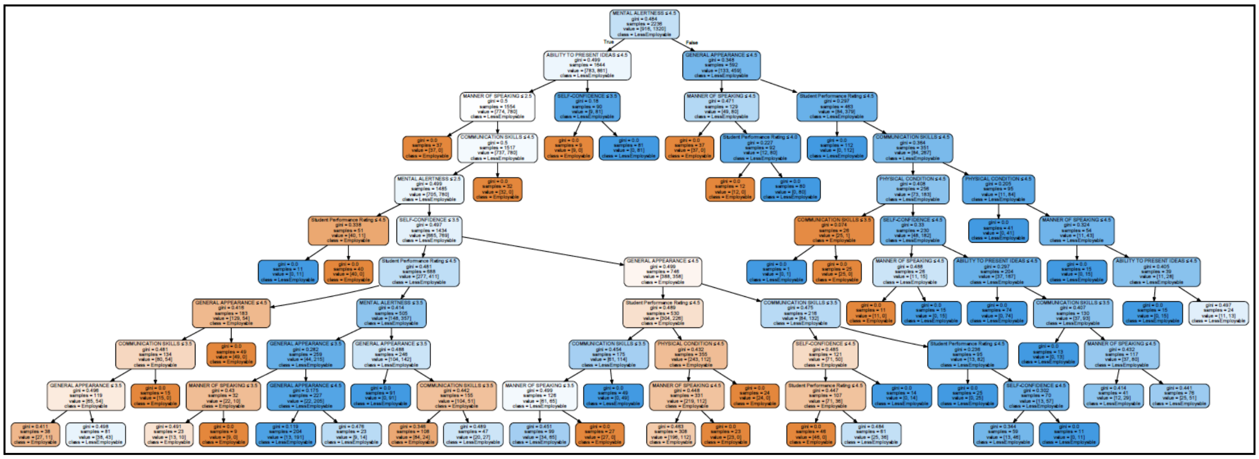

MENTAL ALERTNESS ≤ 4.5 gini = 0.484 samples = 2236 value = [916, 1320] class = LessEmployable
True
False
ABILITY TO PRESENT IDEAS ≤ 4.5 gini = 0.499 samples = 1644 value = [783, 861] class = LessEmployable
GENERAL APPEARANCE ≤ 4.5 gini = 0.348 samples = 592 value = [133, 459] class = LessEmployable
MANNER OF SPEAKING ≤ 2.5 gini = 0.5 samples = 1554 value = [774, 780] class = LessEmployable
SELF-CONFIDENCE ≤ 3.5 gini = 0.18 samples = 90 value = [9, 81] class = LessEmployable
MANNER OF SPEAKING ≤ 4.5 gini = 0.471 samples = 129 value = [49, 80] class = LessEmployable
Student Performance Rating ≤ 4.5 gini = 0.297 samples = 463 value = [84, 379] class = LessEmployable
gini = 0.0 samples = 37 value = [37, 0] class = Employable
COMMUNICATION SKILLS ≤ 4.5 gini = 0.5 samples = 1517 value = [737, 780] class = LessEmployable
gini = 0.0 samples = 9 value = [9, 0] class = Employable
gini = 0.0 samples = 81 value = [0, 81] class = LessEmployable
gini = 0.0 samples = 37 value = [37, 0] class = Employable
Student Performance Rating ≤ 4.0 gini = 0.227 samples = 92 value = [12, 80] class = LessEmployable
gini = 0.0 samples = 112 value = [0, 112] class = LessEmployable
COMMUNICATION SKILLS ≤ 4.5 gini = 0.364 samples = 351 value = [84, 267] class = LessEmployable
MENTAL ALERTNESS ≤ 2.5 gini = 0.499 samples = 1485 value = [705, 780] class = LessEmployable
gini = 0.0 samples = 32 value = [32, 0] class = Employable
gini = 0.0 samples = 12 value = [12, 0] class = Employable
gini = 0.0 samples = 80 value = [0, 80] class = LessEmployable
PHYSICAL CONDITION ≤ 4.5 gini = 0.408 samples = 256 value = [73, 183] class = LessEmployable
PHYSICAL CONDITION ≤ 4.5 gini = 0.205 samples = 95 value = [11, 84] class = LessEmployable
Student Performance Rating ≤ 4.5 gini = 0.338 samples = 51 value = [40, 11] class = Employable
SELF-CONFIDENCE ≤ 3.5 gini = 0.497 samples = 1434 value = [665, 769] class = LessEmployable
COMMUNICATION SKILLS ≤ 3.5 gini = 0.074 samples = 26 value = [25, 1] class = Employable
SELF-CONFIDENCE ≤ 4.5 gini = 0.33 samples = 230 value = [48, 182] class = LessEmployable
gini = 0.0 samples = 41 value = [0, 41] class = LessEmployable
MANNER OF SPEAKING ≤ 4.5 gini = 0.324 samples = 54 value = [11, 43] class = LessEmployable
gini = 0.0 samples = 11 value = [0, 11] class = LessEmployable
gini = 0.0 samples = 40 value = [40, 0] class = Employable
Student Performance Rating ≤ 4.5 gini = 0.481 samples = 688 value = [277, 411] class = LessEmployable
GENERAL APPEARANCE ≤ 4.5 gini = 0.499 samples = 746 value = [388, 358] class = Employable
gini = 0.0 samples = 1 value = [0, 1] class = LessEmployable
gini = 0.0 samples = 25 value = [25, 0] class = Employable
MANNER OF SPEAKING ≤ 4.5 gini = 0.488 samples = 26 value = [11, 15] class = LessEmployable
ABILITY TO PRESENT IDEAS ≤ 4.5 gini = 0.297 samples = 204 value = [37, 167] class = LessEmployable
gini = 0.0 samples = 15 value = [0, 15] class = LessEmployable
ABILITY TO PRESENT IDEAS ≤ 4.5 gini = 0.405 samples = 39 value = [11, 28] class = LessEmployable
GENERAL APPEARANCE ≤ 4.5 gini = 0.416 samples = 183 value = [129, 54] class = Employable
MENTAL ALERTNESS ≤ 3.5 gini = 0.414 samples = 505 value = [148, 357] class = LessEmployable
Student Performance Rating ≤ 4.5 gini = 0.489 samples = 530 value = [304, 226] class = Employable
COMMUNICATION SKILLS ≤ 3.5 gini = 0.475 samples = 216 value = [84, 132] class = LessEmployable
gini = 0.0 samples = 11 value = [11, 0] class = Employable
gini = 0.0 samples = 15 value = [0, 15] class = LessEmployable
gini = 0.0 samples = 74 value = [0, 74] class = LessEmployable
COMMUNICATION SKILLS ≤ 3.5 gini = 0.407 samples = 130 value = [37, 93] class = LessEmployable
gini = 0.0 samples = 15 value = [0, 15] class = LessEmployable
gini = 0.497 samples = 24 value = [11, 13] class = LessEmployable
COMMUNICATION SKILLS ≤ 3.5 gini = 0.481 samples = 134 value = [80, 54] class = Employable
gini = 0.0 samples = 49 value = [49, 0] class = Employable
GENERAL APPEARANCE ≤ 3.5 gini = 0.282 samples = 259 value = [44, 215] class = LessEmployable
GENERAL APPEARANCE ≤ 3.5 gini = 0.488 samples = 246 value = [104, 142] class = LessEmployable
COMMUNICATION SKILLS ≤ 3.5 gini = 0.454 samples = 175 value = [61, 114] class = LessEmployable
PHYSICAL CONDITION ≤ 4.5 gini = 0.432 samples = 355 value = [243, 112] class = Employable
SELF-CONFIDENCE ≤ 4.5 gini = 0.485 samples = 121 value = [71, 50] class = Employable
Student Performance Rating ≤ 4.5 gini = 0.236 samples = 95 value = [13, 82] class = LessEmployable
gini = 0.0 samples = 13 value = [0, 13] class = LessEmployable
MANNER OF SPEAKING ≤ 4.5 gini = 0.432 samples = 117 value = [37, 80] class = LessEmployable
GENERAL APPEARANCE ≤ 3.5 gini = 0.496 samples = 119 value = [65, 54] class = Employable
gini = 0.0 samples = 15 value = [15, 0] class = Employable
MANNER OF SPEAKING ≤ 3.5 gini = 0.43 samples = 32 value = [22, 10] class = Employable
GENERAL APPEARANCE ≤ 4.5 gini = 0.175 samples = 227 value = [22, 205] class = LessEmployable
gini = 0.0 samples = 91 value = [0, 91] class = LessEmployable
COMMUNICATION SKILLS ≤ 3.5 gini = 0.442 samples = 155 value = [104, 51] class = Employable
MANNER OF SPEAKING ≤ 3.5 gini = 0.499 samples = 126 value = [61, 65] class = LessEmployable
gini = 0.0 samples = 49 value = [0, 49] class = LessEmployable
MANNER OF SPEAKING ≤ 4.5 gini = 0.448 samples = 331 value = [219, 112] class = Employable
gini = 0.0 samples = 24 value = [24, 0] class = Employable
Student Performance Rating ≤ 4.5 gini = 0.447 samples = 107 value = [71, 36] class = Employable
gini = 0.0 samples = 14 value = [0, 14] class = LessEmployable
gini = 0.0 samples = 25 value = [0, 25] class = LessEmployable
SELF-CONFIDENCE ≤ 4.5 gini = 0.302 samples = 70 value = [13, 57] class = LessEmployable
gini = 0.414 samples = 41 value = [12, 29] class = LessEmployable
gini = 0.441 samples = 76 value = [25, 51] class = LessEmployable
gini = 0.411 samples = 38 value = [27, 11] class = Employable
gini = 0.498 samples = 81 value = [38, 43] class = LessEmployable
gini = 0.491 samples = 23 value = [13, 10] class = Employable
gini = 0.0 samples = 9 value = [9, 0] class = Employable
gini = 0.119 samples = 204 value = [13, 191] class = LessEmployable
gini = 0.476 samples = 23 value = [9, 14] class = LessEmployable
gini = 0.346 samples = 108 value = [84, 24] class = Employable
gini = 0.489 samples = 47 value = [20, 27] class = LessEmployable
gini = 0.451 samples = 99 value = [34, 65] class = LessEmployable
gini = 0.0 samples = 27 value = [27, 0] class = Employable
gini = 0.463 samples = 308 value = [196, 112] class = Employable
gini = 0.0 samples = 23 value = [23, 0] class = Employable
gini = 0.0 samples = 46 value = [46, 0] class = Employable
gini = 0.484 samples = 61 value = [25, 36] class = LessEmployable
gini = 0.344 samples = 59 value = [13, 46] class = LessEmployable
gini = 0.0 samples = 11 value = [0, 11] class = LessEmployable

Figure 10. Decision Tree Graph (Link here for a clearer image)

Show in Figure 10 is the Decision Tree Graph. Each node represents a decision based on one of the input features and each branch represents a possible outcome of that decision. The components in the Decision Tree graph are: Gini, Samples, Value, and Class. The Gini or the Gini impurity measures how impure a node is or how much the labels in that node are mixed. If the Gini impurity is 0, it means that all the labels in that node are the sample. Samples are the number of samples in that node or the number of data points that fall into that particular combination of feature values. The value of a node is a tuple that shows the number of samples for each class label that fall into that node. If there are 20 samples in a node and 15 of them belong to class 1, the value would be [15, 5]. Lastly, the class is the predicted class for that node; it is the class that most of the samples in that node belong to.

To interpret the graph, follow the branches from the root node to a leaf node to see how the input features are used to make a prediction. The values of the components in each node along the way will show how the model is making its predictions. As the tree has a depth of 10, it is difficult for the group to explain the rules created by the tree. However, In correlation to the feature importance in Figure 9, it seems to be correct that the important feature like “Student Performance Rating”, plays a big role in the decision-making of the tree and is the same for the other, similarly important features.

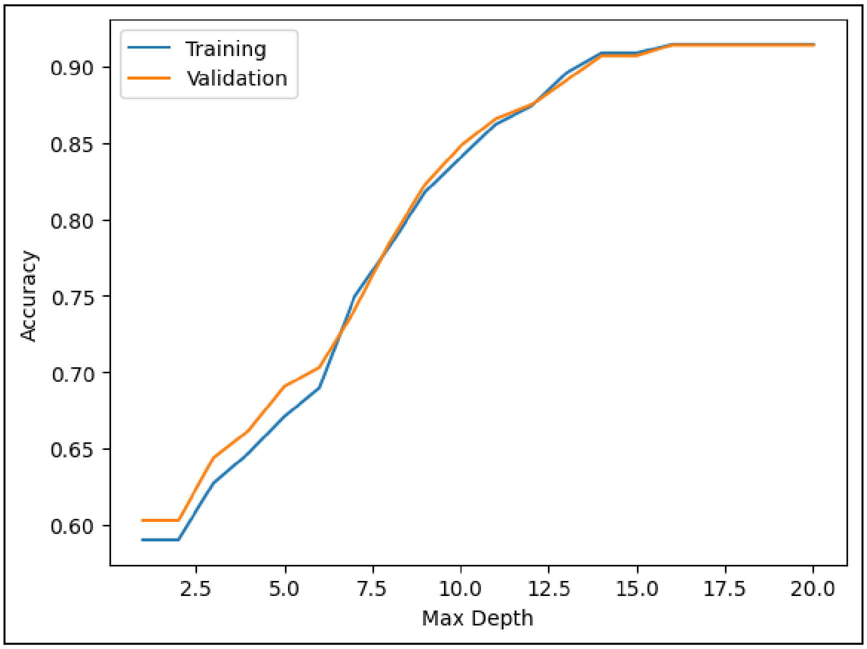


Figure 11. Decision Tree Validation Curve

In Machine Learning, it is important to measure the effectiveness of the model through validation (Pramoditha, 2021). In the figure above, the group is trying to identify if the model is an underfit, overfit, or if it is a good fit. An underfit model occurs when the model is not properly trained, this can result in the model performing poorly on new data. On the other hand, overfit data occurs when the model is overly complex. This can result in a model that misclassifies new data (Muralidhar, 2021). Lastly, a well-trained model (or a good fit), is identified when the training and validation loss decreases and stabilizes with a minimal gap between the two (Browenlee, 2019).

In the graph above, It can be seen that the curve has minimal gaps at the first few depths of the tree (< 7.5 Max depth). This gap continuously decreases as the decision tree traverses deeper (< 15 Max depth). The curve reached a plateau and stabilized more towards the end (< 20 max depth). Given this graph, it perfectly fits the description of a model with a good fit.

It can also be said that since the gap between training and validation lines is minimal across all depths, the group could increase the max depth parameter to increase the model's performance. However, this might just be a coincidence for this specific dataset.

Given the in-depth analysis of the performance of both models. It is important to note that these tests are limited to a single train-test split and should not be taken at face value. To help verify that the performance of each model is similar outside the single train-test split, the concept called cross-validation is done. According to Sharma (2023), cross-validation is a machine learning technique used to evaluate a model's performance on unseen data. Cross-validation is important in machine learning because, in actual practice, machine learning models have to work on real-world, dynamic data which the model might struggle if the training data it trained itself on contained too much noise, or as tackled previously, overfitted or underfitted the dataset on its learning. In cross-validation, the implementation involves dividing the available data into multiple folds or subsets, where one of the folds is used as the validation set. This process is repeated multiple times and the result from each validation step is averaged to produce a robust estimate of the model's performance (Sharma, 2023).

There are many variations of the cross-validation technique, but the group decided to use the K-fold cross-validation variation. According to Vadapalli (2022), the data is divided into k subsets wherein each division, one of the k subsets is used as the validation set, and the other $k - 1$ subsets as the training set. The result of each k trial will then be averaged to get an accurate and efficient performance estimate of a model.

Fortunately, sci-kit gives data scientists an easy, abstracted implementation of cross-validation techniques like K-Fold cross-validation. In sci-kit, the implementation is a function called *"cross_val_score."* This function, as discussed, evaluates a score by using cross-validation. There are many hyperparameters to control this function. Still, the most important is the number of K-folds the technique will do. According to Olsen (2023), the number of K-folds cannot be optimally defined. However, it is important not to go overboard with the number of K-folds as it can lead to overfitting of data. Given that the dataset used in the problem is relatively small, the group decided to use 10 K-folds, which seems appropriate for a dataset size such as the one used and is also a common number of K-fold.

To visualize the cross-validation score result, a box plot graph will be done.

Box plot graphs were used to represent cross-validation score results because it is a concise and easier way to interpret them. This is because it can easily show the distribution of the scores, central tendency, and the data spread. This can help quickly evaluate the model's robustness given the result of the cross-validation technique. Thus, below is a box plot graph of the cross-validation result for Logistic Regression.

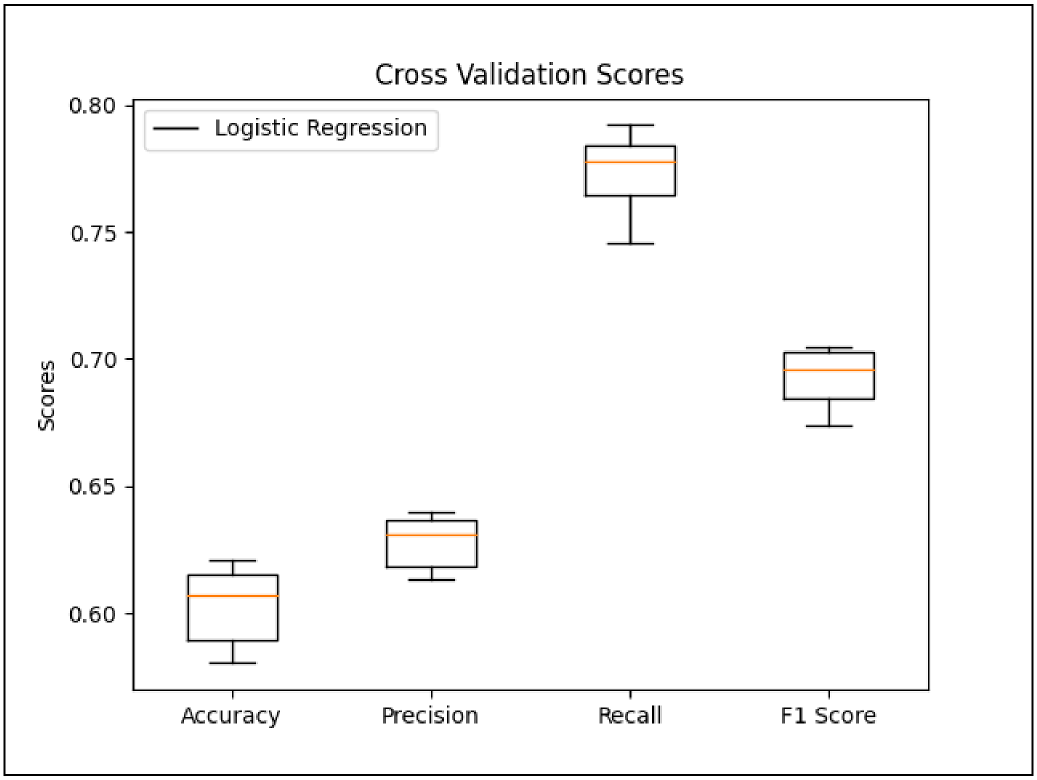


Figure 12. Box Plot Graph of the Cross-Validation Score of Each Metric For Logistic Regression

Shown in Figure 12, is the result of each 10-fold cross-validation test result for Logistic Regression. In the graph, the result for each metric is approximately similar to the single train-test split. But to fully know, data needs to be retrieved for analysis. Multiple data can be extracted to delve into the results thoroughly. Still, for simplicity, only the mean and the standard deviation will be tabulated to show the average score across all 10-fold tests easily and also to know how spread apart the scores are to the mean or center. Thus, below is the tabulation of the cross-validation score result.

Table 3. Cross-Validation 10-Fold Result Score for Each Metric of Logistic Regression

METRIC	MEAN	STANDARD DEVIATION
ACCURACY	0.602943	0.015002
PRECISION	0.62811165797432	0.010033
RECALL	0.772708	0.014526
F1-SCORE	0.692935	0.011641

Seen in Table 3, the score is relatively similar to the scores in Table 2, however, to easily visualize the comparison, below is a bar graph to compare the scores.

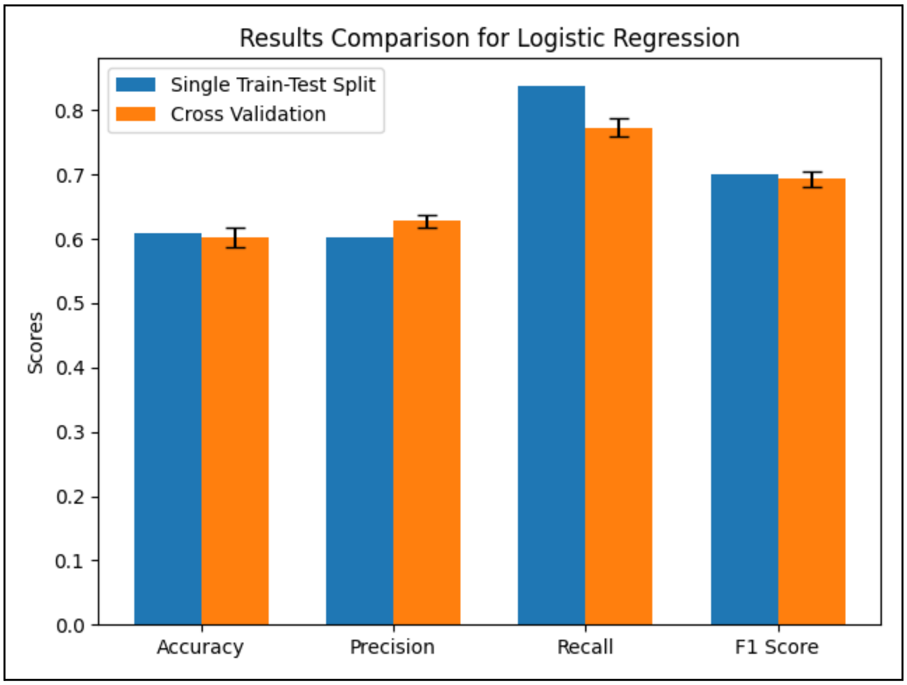


Figure 13. Results Comparison for Logistic Regression

As Seen in Figure 13, the average score of the 10-fold test result is similar to the single train-test split result. There is, however, a slight difference in the average score, the most notable one is the average score for recall wherein the single train-test split result is interpreted to be “Great” performance, but now it is in the range of 60-70%, which is interpreted as “Good.” Nevertheless, given the similar results, it can be concluded that in general, the Logistic Regression model is able to handle new, unseen data outside its training-test dataset. Although its performance across the metrics ranges from “Ok” to “Good,” it is still usable as a way to predict students’ employability but it is important to note that there might still be ways to improve the model’s performance that was not touched upon by the group

Moving to the next model, seen below, is the box plot 10-fold cross-validation test result of each metric for the Decision Tree model.

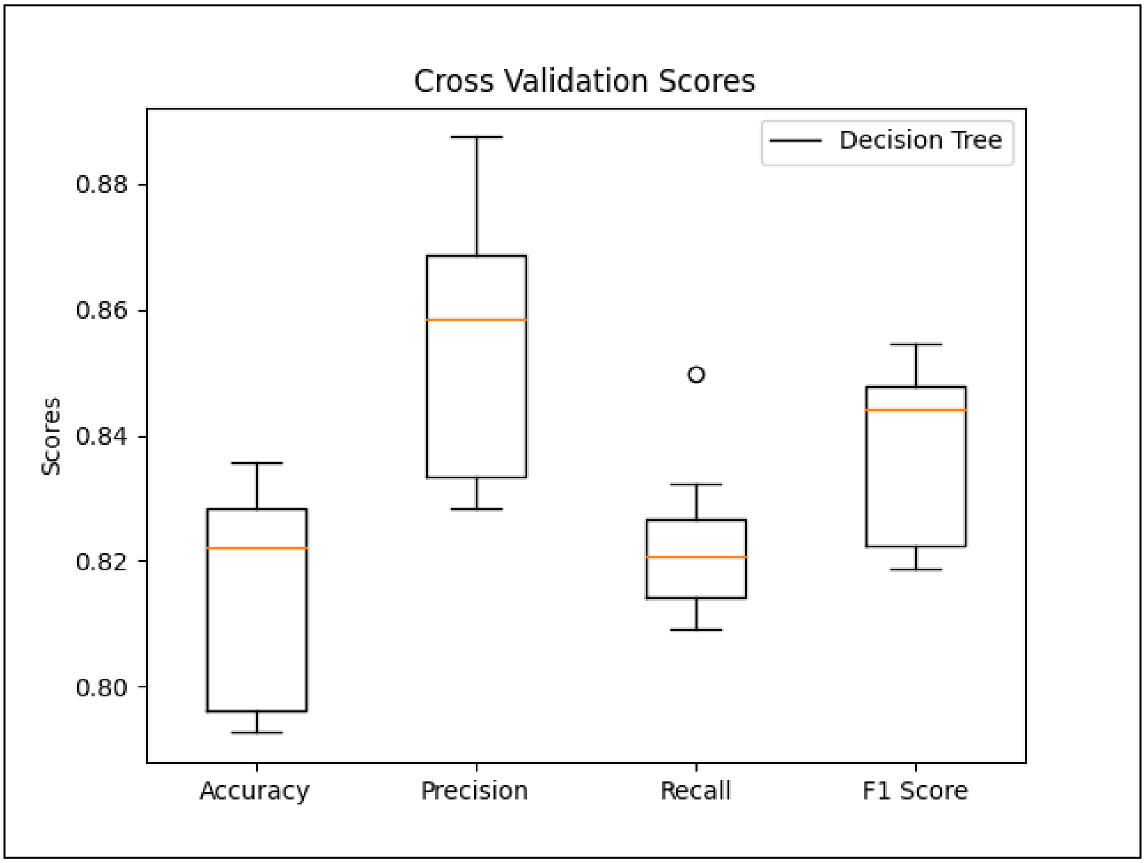


Figure 14. Box Plot Graph of the Cross-Validation Score of Each Metric For Decision Tree

Seen in Figure 14, the 10-fold cross-validation result for each metric seems to have a similar result to its single train-test split result. There is, however, more data that can be interpreted in the graph. The first notable is the whisker in precision, it seems to have an unusually long upper whisker from the third quartile (Q3). This suggests that there is a lot of variability in the scores in that range. The second notable is the presence of an outlier in the recall metric. This suggests that there was an unusually high score result during its K-fold phase. Like Logistic Regression, a lot of data can be extracted from the box plot graph. Still, only the mean and standard deviation will be discussed further for simplicity. Below is the tabulation of the mean and standard deviation for each metric of the Decision Tree model.

Table 4. Cross-Validation 10-Fold Result Score for Each Metric of Decision Tree

METRIC	MEAN	STANDARD DEVIATION
ACCURACY	0.814892	0.01638
PRECISION	0.853334	0.019865
RECALL	0.822436	0.011649
F1-SCORE	0.837515	0.013618

Seen in Table 4, the score is relatively similar to the scores in Table 2, however, to easily visualize the comparison, below is a bar graph to compare the scores.

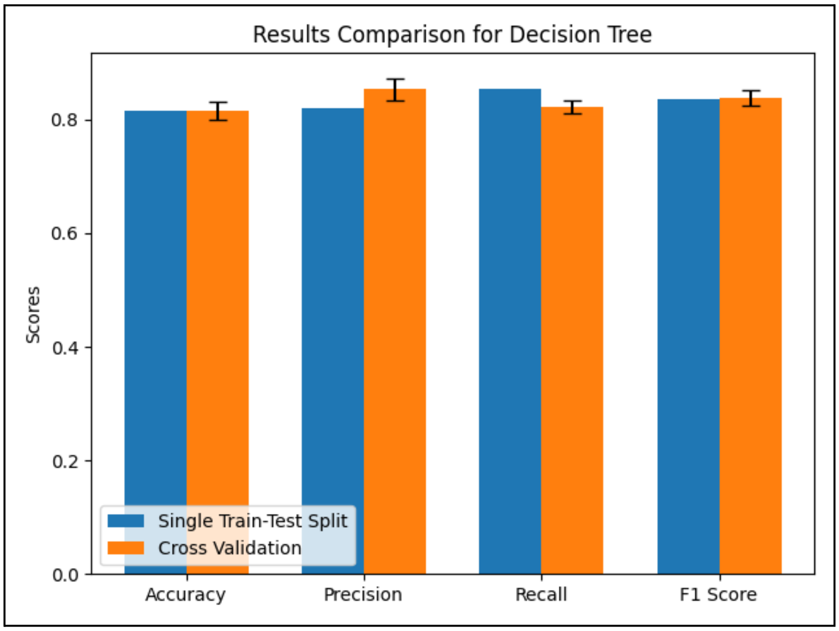


Figure 15. Results Comparison for Decision Tree

As seen in Figure 15, the visualization helps to suggest that the single train-test score result is similar to the average score of the 10-fold cross-validation result. As seen in Table 4, the average score can still be interpreted as “Great,” this means that the Decision Tree model can generally handle new, unseen data very well. Given its performance in each metric, it can be said that it is a superior model to use in predicting students’ employability compared to Logistic Regression. Nonetheless, there are still various ways to improve the Decision Tree’s performance that was not touched upon by the group.

In summary, given the in-depth analysis and the validation of the analysis of performance for both models, it can be concluded that the Decision Tree worked better for this specific binary classification problem because it was able to capture the important relationships between the independent features or the various attributes of a student and the target variable or the students’ employability. Meanwhile, Logistic Regression, to an extent, was still able to capture some relationships, but its limitation made it perform worse. Thus, the group was able to showcase both models’ performance, strengths, and limitations in solving a binary classification problem.

V. Conclusions and Recommendations

The group was tasked with understanding machine learning algorithms by simulating at least two models on an eligible dataset. The chosen dataset was the "Students’ Employability Dataset - Philippines" from Kaggle, which has 8 independent features rating different attributes of a student, such as "Mental Awareness," and a binary target variable labeled "Employable" or "Less Employable." The dataset has a sample size of 2982, making it suitable for a simple machine learning performance analysis. The group chose Logistic Regression and Decision Tree as the models to use because they are easy to interpret. However, no preprocessing was done on the dataset, only encoding the label classes into a readable format by turning "Employable" into 1 and "Less Employable" into 0, which means a positive and a negative outcome respectively. The group changed only the hyperparameters, including the max iteration for Logistic Regression, max depth for Decision Tree, and k-folds for cross-validation. After training, analyzing, and validating the models' performance using appropriate metrics, the group concluded that Decision Tree performed better than Logistic Regression, as it captured the relationship between the independent features and the target variable more effectively than Logistic Regression, which only captured a part of the relationship.

As this is a simple machine learning analysis done by the group, there are various recommendations that can be suggested to improve the task of evaluating and improving the performance of machine learning models. The group first recommends that the sample size of the data set should be larger. Although the data set used had a reasonable sample size, having a larger size means there are more data for the model to train itself and might be able to capture more information that can improve the prediction performance.

In tandem with the previous recommendation, the group also recommends using appropriate preprocessing techniques. This is because, as the dataset grows larger, there is more potential for outliers or data that can affect the performance of the model. There are multitudes of preprocessing techniques to choose like normalization, standardization, and feature selection. But usually, it involves the process of making sure the data is cleaned, readable, and scaled correctly for the machine to properly learn and train from.

The group also recommends properly configuring the hyperparameters used in various functions in sci-kit. The group briefly explained these hyperparameters but was not expounded upon. However, these hyperparameters are important as the right combination can yield significant results. For Logistic Regression, there are more hyperparameters besides max iterations, like the regularization parameter, which helps prevent overfitting training data by penalizing large coefficients. For Decision Tree, some hyperparameters can control the number of splits and leaf nodes in the tree, which also helps prevent overfitting training data. By finding the best combination of hyperparameters, the prediction performance can be improved.

Additionally, the group recommends conducting a deeper analysis of Logistic Regression and Decision Tree as there is more to be analyzed than what the group touched upon. Although Logistic Regression was found to perform worse than Decision Tree in the analysis, there are still other metrics, such as the ROC curve and AUC, that can be used to further evaluate how well the model predicts positive and negative outcomes. A better understanding of how the model trains the data, such as how the logistic and cost functions work, can also provide deeper insights into the coefficients in Logistic Regression. Similarly, for Decision Tree, although it performed better and was verified to not be overfitting the data, there are still intricate and interesting aspects of the algorithm, such as Gini impurity and entropy importance, that were touched upon but not fully defined.

Lastly, the group recommends exploring more classification algorithms. In the world of machine learning, where AI is prevalent in today’s society, various algorithms exist that may work better than the ones the group touched upon. Some notable classification algorithms include Random Forest, which combines multiple decision trees to improve performance; Naive Bayes, a model that makes predictions based on the probability of a given feature belonging to a certain class; and Neural Networks, which use layers of interconnected nodes to learn complex patterns in the data to predict classifications. However, the group adds that it’s important to choose the appropriate algorithm for the problem at hand. Simple problems may only require simple algorithms, as this can help with interpreting the results and avoiding long processing times due to the complexity of some algorithms.

VI. References

Agarwal, R. (2019, September 18). The 5 Classification Evaluation Metrics You Must Know. Towards Data Science. Retrieved from https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226

Agrawal, S. K. (2022, December 2). Metrics to Evaluate your Classification Model to take the right decisions. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

Anas Hamoutni. Students Employability Dataset, Version 1.0. Kaggle, 2021. Accessed on April 7, 2023. https://www.kaggle.com/datasets/anashamoutni/students-employability-dataset.

Bhandari, P. (2022, November 17). Ordinal data: Definition, examples, key characteristics. Scribbr. https://www.scribbr.com/statistics/ordinal-data/

Bujokas, E. (2022, June 2). *Feature importance in Decision Trees*. Medium. Retrieved April 12, 2023, from https://towardsdatascience.com/feature-importance-in-decision-trees-e9450120b445

Brownlee, J. (2019, August 6). How to use Learning Curves to Diagnose Machine Learning Performance. Retrieved on April 14, 2023, from https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/#:~:text=A%20good%20fit%20is%20identified,dataset%20than%20the%20validation%20dataset.

Chauhan, N. S. (2022). *Decision Tree algorithm, explained*. KDnuggets. Retrieved April 12, 2023, from https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

Dawson, E. (2022, September 15). *What is employability? 7 skills for being attractive to employers*. Future Fit. Retrieved April 7, 2023, from https://www.futurefit.co.uk/blog/what-is-employability/

Fajaryati, N., Budiyono, Akhyar, M., & Wiranto. (2020, January 1). *The employability skills needed to face the demands of work in the future: Systematic literature reviews*. De Gruyter. Retrieved April 8, 2023, from https://www.degruyter.com/document/doi/10.1515/eng-2020-0072/html?lang=en

Glass Box Medicine. (2019). Confusion matrix [Digital image]. Retrieved from https://glassboxmedicine.files.wordpress.com/2019/02/confusion-matrix.png?w=1024

Google Developers. (2022, July 18). Precision and recall. Retrieved from https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

Gupta, P. (2017, November 12). *Decision Trees in machine learning*. Medium. Retrieved April 11, 2023, from https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

Hosain, M. S., Mustafi, M. A., & Parvin, T. (2021). Factors affecting the employability of private university graduates: An exploratory study on Bangladeshi employers. *PSU Research Review*. https://doi.org/10.1108/prr-01-2021-0005

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*(4), 531–538. https://doi.org/10.1002/sam.11583

Kelley, K. (2023, March 30). *Why are employability skills important?: Simplilearn*. Simplilearn.com. Retrieved April 7, 2023, from https://www.simplilearn.com/why-are-employability-skills-important-article

Kidd, C. (2022, October 28). *Data Normalization explained: How to normalize data*. Splunk. Retrieved April 11, 2023, from https://www.splunk.com/en_us/blog/learn/data-normalization.html

Kisselev, D. (2021, September 16). A simple interpretation of logistic regression coefficients. Towards Data Science. https://towardsdatascience.com/a-simple-interpretation-of-logistic-regression-coefficients-e3a40a62e8cf

Korstanje, J. (2021, August 31). The F1 Score. Towards Data Science. Retrieved from https://towardsdatascience.com/the-f1-score-bec2bbc38aa6

Muralidhar, K. (2021, February 9). Learning Curve to identify Overfitting and Underfitting in Machine Learning. Retrieved April 14, 2023, from https://towardsdatascience.com/learning-curve-to-identify-overfitting-underfitting-problems-133177f38df5#:~:text=Learning%20curve%20of%20a%20good%20fit%20model%20has%20a%20high,model%20performance%20on%20unseen%20data.

Narkhede, S. (2018, May 9). Understanding Confusion Matrix. Towards Data Science. Retrieved from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Olsen, L. R. (2023, January 26). Picking the number of folds for cross-validation. The Comprehensive R Archive Network. https://cran.r-project.org/web/packages/cvms/vignettes/picking_the_number_of_folds_for_cross-validation.html

Patel, H. (2021, September 2). *What is feature engineering-importance, tools and techniques for machine learning*. Medium. Retrieved April 11, 2023, from https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10

Pramoditha, R. (2021, March 13). Validation Curve Explained –Plot the influence of a single hyperparameter. Retrieved April 14, 2023, from https://towardsdatascience.com/validation-curve-explained-plot-the-influence-of-a-single-hyperparameter-1ac4864deaf8#:~:text=The%20validation%20curve%20is%20a,some%20range%20of%20hyperparameter%20values.

Raj, A. (2021, January 5). *The perfect recipe for classification using logistic regression*. Medium. Retrieved April 12, 2023, from https://towardsdatascience.com/the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592#:~:text=Logistic%20Regression%20is%20one%20of,fast%20at%20classifying%20unknown%20records.

Rouse, M. (2019, August 2). *Scikit-Learn*. Techopedia. Retrieved April 8, 2023, from https://www.techopedia.com/definition/33860/scikit-learn

Rowe, W., & Johnson, J. (2020, September 8). *Top machine learning frameworks to use*. BMC Blogs. Retrieved April 8, 2023, from https://www.bmc.com/blogs/machine-learning-ai-frameworks/

Roy, A. (2020, November 6). *A dive into Decision Trees*. Medium. Retrieved April 12, 2023, from https://towardsdatascience.com/a-dive-into-decision-trees-a128923c9298

Ruble, R. A. (2018, December 19). *The sage encyclopedia of communication research methods*. Sage Research Methods. Retrieved April 7, 2023, from https://methods.sagepub.com/reference/the-sage-encyclopedia-of-communication-research-methods/i3002.xml#:~:text=A%20communication%20skill%20is%20defined,engages%20in%20particular%20communication%20behaviors.

Scikit-learn developers. (n.d.). cross_val_score. scikit-learn. Retrieved April 7, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

Scikit-learn developers. (n.d.). Logistic regression. scikit-learn. Retrieved April 7, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Scikit-learn developers. (n.d.). Model evaluation: quantifying the quality of predictions. scikit-learn. Retrieved April 7, 2023, from https://scikit-learn.org/stable/modules/model_evaluation.html

Sharma, A. (2023, February 15). Cross Validation in Machine Learning. GeeksforGeeks. https://www.geeksforgeeks.org/cross-validation-machine-learning/

Tadese, M., Yeshaneh, A., & Mulu, G. B. (2022). Determinants of good academic performance among university students in Ethiopia: A cross-sectional study. *BMC Medical Education*, *22*(1). https://doi.org/10.1186/s12909-022-03461-0

University College London. (2021, April 15). *How personality affects work behaviour and career success*. UCL. Retrieved April 7, 2023, from https://onlinelearning.ucl.ac.uk/resources/personality-affects-work-behaviour-career-success/

Vadapalli, P. (2022, September 1). Cross Validation in Machine Learning: 4 Types of Cross Validation. Upgrad. https://www.upgrad.com/blog/cross-validation-in-machine-learning/

Wolff, R. (2020, August 27). Classification Algorithms: The Complete Guide. MonkeyLearn Blog. Retrieved from https://monkeylearn.com/blog/classification-algorithms/

Yadav, D. (2019, December 9). *Categorical encoding using label-encoding and one-hot-encoder*. Medium. Retrieved April 11, 2023, from https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd

Zach. (2021, August 4). Why is Standard Deviation Important? Statology. https://www.statology.org/why-is-standard-deviation-important/

Zach. (2022, February 18). The Importance of the Mean in Statistics. Statology. https://www.statology.org/importance-of-mean/

Zach. (2022, May 19). What is Considered a Good Accuracy for a Machine Learning Model? Statology. https://www.statology.org/good-accuracy-machine-learning

VII. Contributions of Each Member

Contribution Table

Names	Contributions
Balderosa, Ernest	● Helped contribute to the introduction ● Helped contribute to the methodology of the report ● Helped contribute to the decision tree validation analysis of the report
Caasi, Samantha Nicole	● Contributed to the introduction part of the report ● Contributed to the data set part of the report ● Contributed to the methodology of the report ● Very minimal contribution to the results and analysis part of the report
Marcellana, John Patrick	● Helped contribute to the introduction of the report ● Helped contribute to the methodology of the report ● Helped contribute to the results and analysis of the report

Noche, Zach Matthew	● Helped contribute to the code setup ● Contributed to the dataset analysis ● Contributed to the results and analysis ● Contributed to the recommendation