**Table of Contents**

[Part 1 4](#_Toc115104841)

[Q1. Explain the chosen online learning behavior indicator and dimension. 4](#_Toc115104842)

[Q2. Design an architecture framework for proposed model 6](#_Toc115104843)

[Q3. Overall Design of architecture framework 7](#_Toc115104844)

[Part 2 12](#_Toc115104845)

[Q1. Software to implement Artificial Intelligence 12](#_Toc115104846)

[Q2. Plotting Experimental Data 12](#_Toc115104847)

[K-Nearest Neighbors (KNN) Model 12](#_Toc115104848)

[Decision Tree Model 14](#_Toc115104849)

[Support Vector Machine (SVM) Model 16](#_Toc115104850)

[Logistic Regression Model 18](#_Toc115104851)

[Q3. Software features, results, and critical explanations 20](#_Toc115104852)

[Import libraries to training 20](#_Toc115104853)

[Model Implementation 23](#_Toc115104854)

[Result Explanation 27](#_Toc115104855)

[Model Evaluation 29](#_Toc115104856)

[References 37](#_Toc115104857)

**Table of Figures**

[Figure 1 Behaviour Indicators Selection 6](#_Toc112406478)

[Figure 2 Model Framework Architecture 7](#_Toc112406479)

[Figure 3 Comparison Table of Algorithms 8](#_Toc112406480)

[Figure 4 KNN Algorithm (Javatpoint, 2021) 9](#_Toc112406481)

[Figure 5 Decision Tree Classification Algorithm (Javatpoint, 2021) 10](#_Toc112406482)

[Figure 6 SVM Algorithm (Javatpoint, 2021) 11](#_Toc112406483)

[Figure 7 Logistic Regression (Javatpoint, 2021) 12](#_Toc112406484)

[Figure 8 Architecture Framework Flowchart 12](#_Toc112406485)

[Figure 9 Visit resources (KNN) 13](#_Toc112406486)

[Figure 10 Announcement Views (KNN) 13](#_Toc112406487)

[Figure 11 Discussions (KNN) 14](#_Toc112406488)

[Figure 12 Absence (KNN) 14](#_Toc112406489)

[Figure 13 Visit Resources (Decision Tree) 15](#_Toc112406490)

[Figure 14 Announcement Views (Decision Tree) 15](#_Toc112406491)

[Figure 15 Discussions (Decision Tree) 16](#_Toc112406492)

[Figure 16 Absence (Decision Tree) 16](#_Toc112406493)

[Figure 17 Visit Resources (SVM) 17](#_Toc112406494)

[Figure 18 Announcement Views (SVM) 17](#_Toc112406495)

[Figure 19 Discussions (SVM) 18](#_Toc112406496)

[Figure 20 Absence (SVM) 18](#_Toc112406497)

[Figure 21 Visit Resources (Logistic Regression) 19](#_Toc112406498)

[Figure 22 Announcement Views (Logistic Regression) 19](#_Toc112406499)

[Figure 23 Discussions (Logistic Regression) 20](#_Toc112406500)

[Figure 24 Absence (Logistic Regression) 20](#_Toc112406501)

[Figure 25 Library imports for training 21](#_Toc112406502)

[Figure 26 Dataset head 21](#_Toc112406503)

[Figure 27 Selected dataset 21](#_Toc112406504)

[Figure 28 Split the train and test dataset 22](#_Toc112406505)

[Figure 29 Get y dependent variables 22](#_Toc112406506)

[Figure 30 Classify absence 22](#_Toc112406507)

[Figure 31 Pre-processing for X variables 24](#_Toc112406508)

[Figure 32 Finding k value with best accuracy 24](#_Toc112406509)

[Figure 33 Training other models 25](#_Toc112406510)

[Figure 34 Testing and Analysis 26](#_Toc112406511)

[Figure 35 Visit Resource plot 27](#_Toc112406512)

[Figure 36 Absence days visualization 27](#_Toc112406513)

[Figure 37 Plotting KNN Confusion Matrix 30](#_Toc112406514)

[Figure 38 KNN Confusion Matrix 30](#_Toc112406515)

[Figure 39 Plotting Decision Tree Confusion Matrix 31](#_Toc112406516)

[Figure 40 Decision Tree Confusion Matrix 31](#_Toc112406517)

[Figure 41 Plotting SVM Confusion Matrix 32](#_Toc112406518)

[Figure 42 SVM Confusion Matrix 32](#_Toc112406519)

[Figure 43 Plotting Logistic Regression Confusion Matrix 33](#_Toc112406520)

[Figure 44 Logistic Regression Confusion Matrix 33](#_Toc112406521)

[Figure 45 Plotting F1 scores of each Model 34](#_Toc112406522)

[Figure 46 Model F1-Scores Chart 35](#_Toc112406523)

[Figure 47 Visualizing Model Accuracy 36](#_Toc112406524)

[Figure 48 Model Accuracy Chart 37](#_Toc112406525)

# Part 1

## Q1. Explain the chosen online learning behavior indicator and dimension.

In a paper done by Kuzminykh, I. and company (2021), research was done to investigate the relationship between student’s online activity and their academic performance in online education. The study incorporates data on module content engagement from an e-learning platform, which includes the number of Visits of web page with announcements, number of Posts on the Discussion forums, and quantity of Unique/Repeated content web-page visits. This data was gathered from 139 postgraduate students in an online cybersecurity programme. Daily data collection took place. The study evaluated the connection between academic achievement and learner engagement, which includes all of the previous factors mentioned. Using the Pearson correlation function, they can get the output correlation value, which was used to determine degree of correlation. The test results shows that academic achievement in an online learning environment is influenced by both overall and initial involvement in the online environment. Additionally, starting and overall involvement are closely connected for the data samples analyzed, showing a constant rate of study over the course of the module (Kuzminykh, I. et al. 2021).

A paper by Zhang et al. proposed an interpretable online learner’s performance prediction model based on several learning analytics. After intensive research on online learning behaviour indicators and factors, the study narrowed down to 19 different indicators to be used in the research. However, not all indicators chosen may actually affect the result of the student’s performance. Some indicators may not be suitable for the research as the selected database may not have matching data available for the indicator, or a huge majority of the students in the particular courses do not exhibit certain behaviours corresponding to the indicators. Before constructing the model, the researchers first collect data from online learning platforms. The data collected from the databases have inconsistencies, incomplete and abnormal values, therefore, the researchers conducted per processing on the dataset by removal of the missing and irrelevant values and also combining multiple datasets to a single database source. Data cleansing, format conversion and data separation are also used to enhance the representative meaning of the datasets. After preparing the dataset, correlation analysis is done to check if the chosen indicators are related to the results. If the analysis show that the indicator has an insignificant correlation or correlation that are illogical, it will be abandoned. The researchers concluded that number of course login, time of resource watched, time of resource repeated watch and time difference between post and launch. There indicators will be used for the prediction algorithm. (Zhang et al., 2019)

Tarif (2018) investigated the relationship between student attendance and exam performance. In this research, four courses’ attendance and exam results were evaluated. Every student must gain a 75% and above attendance rate for each subject in order to sit for the exam. The collected dataset consists of 75 students with qualifications to attend the exam. As a result, the research showed that the student’s class attendance positively impacts the final exam results. For example, Course A had an average of 92.6% attendance rate, and the students scored a mean of 24 out of 30 marks. On the other hand, Course D had an average of 91% attendance rate, and the students scored a mean of 17.7 out of 30 marks (Fadelelmoula, 2018). It is proved by the research conducted by Althubaiti et al. (2016) that indicated a 0.27 rise in students’ final block grades as the result of a 1.0% increase in lecture attendance. Hence, lecture attendance is an essential element in determining their academic performance.

To conclude, the first literature review showed that the number of times a student has viewed announcements, resources, and posted discussions, plays a role in determining a student’s academic performance, and as such, will be included in our dataset. Subsequently, the second literature review additionally compounded the linkage of number of resource visits to academic performance, while also introducing another factor of the number of times the student has logged into the course, which may also be interpreted as the students presence for classes. In addition, the third literature review studied specifically on the effect on student performance based on their attendance for lectures, which had showed that student attendance is tied with academic performance as well.

|  |  |  |  |
| --- | --- | --- | --- |
| Article | 1 | 2 | 3 |
| Indicators | View announcements, Discussion, Visit Resource | Visit Resource, Number of Course Login (Absence days) | Attendance (Absence days) |

Figure 1 Behaviour Indicators Selection

## Q2. Design an architecture framework for proposed model

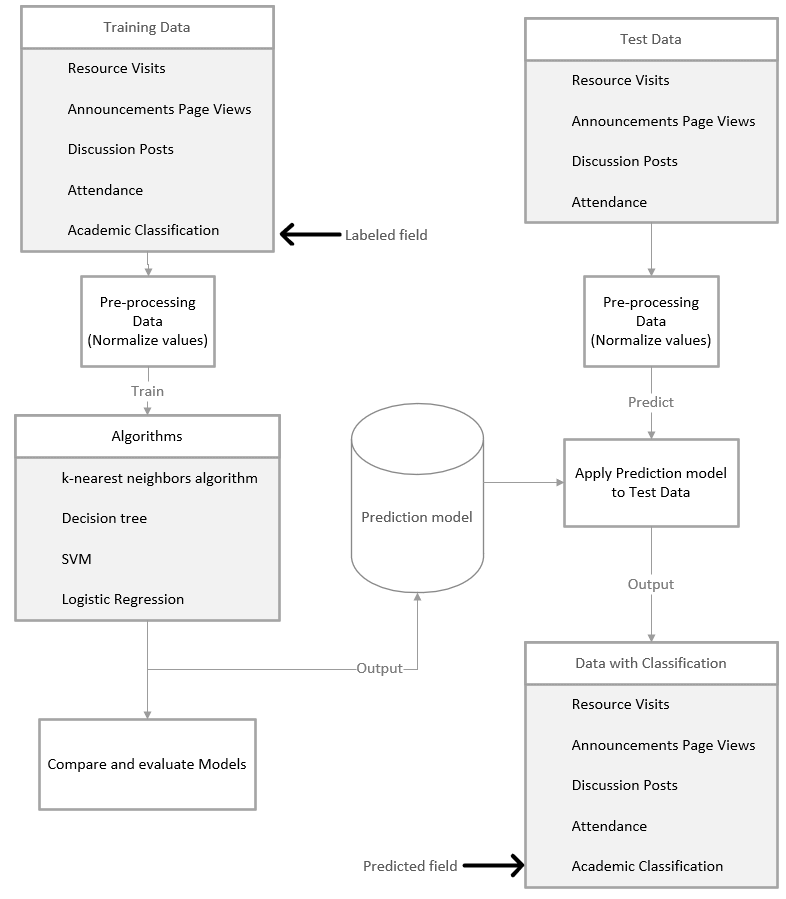


Figure 2 Model Framework Architecture

## Q3. Overall Design of architecture framework

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **K-Nearest Neighbors (KNN)** | **Decision Tree** | **Support Vector Machines (SVM)** | **Logistic Regression** |
| Algorithm Type | Supervised Learning | Supervised Learning | Supervised Learning | Supervised Learning |
| Methods | Classify the data into specific categories based on its proximity by calculating the distance metrics and k values.   * Distance Metrics (Euclidean Distance): Distance between data point and query point of neighbors * K-values: Number of neighbors used to compare the distances | Classify the data into a tree structure.   * Internal Node: Dataset Features * Branch: Decision Rules * Leaf Node: Outcome | Establish a line as the optimal decision boundary that divides n-dimensional space into classes to classify the data into specific categories. | Predict the probability of an event happening. Categorize the data into discrete classes based on the sigmoid function, which accepts negative infinity (-∞) to positive infinity (∞) as input and produces the probability as output. It is utilized while the dependent variable is binary (0/1). |
| Advantages | * Easy to understand and implement * Better performance when there are more complex data * Robust to noisy data | * Does not require feature scaling * Able to deal with the linear and non-linear problem * Better solution for the decision-related problem as it shows all the possible outcomes for the problem | * Robust to overfitting issue * Memory efficiency * Deal better with non-linear problem * A better choice when the relationship between data is unknown | * Easy to understand and implement * Better solution for the linear problem * A better choice when classifying unknown data |
| Disadvantages | * Require the manual input of the number of neighbours (*k* value) * Distance calculation between data points requires high computation cost | * Increased complexity when there are more class labels * Prone to overfitting issue | * Worse performance when there are more complex data * Difficult to understand by the developer | * Prone to overfitting issue when the number of features is more than the number of observations * Not able to deal with the non-linear problem |

Figure 3 Comparison Table of Algorithms

K-Nearest Neighbors (KNN) represent popular implementations under the scope of classifiers and algorithms, with multiple variations of classification ruling being released over the years. Compared to most other machine learning and pattern recognition algorithms, KNN has drawn wide attention thanks to its ease of use, effectiveness, and ease of implementations. This is due to the use of a singular parameter representing *k* and classification resulting from determining majority vote based on the nearest neighbors. This means the sensitivity based on KNN algorithms largely relates back to *k* itself, thus allowing for less reliance and more robustness towards noisy data, and overall a better performing algorithm when it comes to large batches of data (Gou et al., 2019). However, this also acts as a weakness due to the sensitivity of *k*, which the assigned value can have significant effects on both performance and accuracy depending on the sample size. Additionally, distance measuring and calculation require large amounts of computational power to process, though this can be resolved with divide and conquer methods to aid in lowering computational cost and increasing performance (Abu Alfeilat et al., 2019).

Diagram

Description automatically generated with medium confidence

Figure 4 KNN Algorithm (Javatpoint, 2021)

Decision trees are another popular classification algorithm due to the boolean outcomes available for each node and relative ease in visualization of outcomes regarding the approach used and path the algorithm takes. In the sense of classification, it retains popularity because of the ease in writing classification ruling, and not numerical weightage that other neural networks use. As such, decision trees are typically used for grouping purposes and not in-depth classification due to their outcome-based approach (Charbuty & Abdulazeez, 2021). The decision tree used also does not require feature scaling, precisely due to the outcome-based approach making them insensitive to data noise and variances. Additionally, application with discrete data allows solving for both linear and non-linear, whereelse it would only be possible to discern non-linear problems. However, decision trees do suffer from increased complexity and run times with increases in class labels as trees will not always be balanced. Besides that, it is possible for overfitting to occur, which involves over-generation of the tree which causes a drop in generalization of the data (Patel & Prajapati, 2018).



Figure 5 Decision Tree Classification Algorithm (Javatpoint, 2021)

Support vector machines (SVM) have comparatively high generalization capabilities leading to better dealings with non-linear problems, with significant advances on compared algorithms due to the efficiency of memory usage during query and runtime (Cervantes et al., 2020). SVMs are widely used in classification and regression, with the robustness to overfitting transferred from model fitting to model selection a key factor in use of the algorithm, thanks to its use of regularization. SVM is a better algorithm for use when the relationships between data are unknown, due to the margin of classification and catering of misclassification errors in regards to features (Pisner & Schnyer, 2019). However, SVM does not consider noisy or missing data (Pant & Trafalis, 2015). Performance regarding SVM also suffers when complex data is introduced as the initial training is more resource intensive compared to other algorithms. Moreover, SVM is significantly harder to understand and implement due to the complexity of the algorithm.

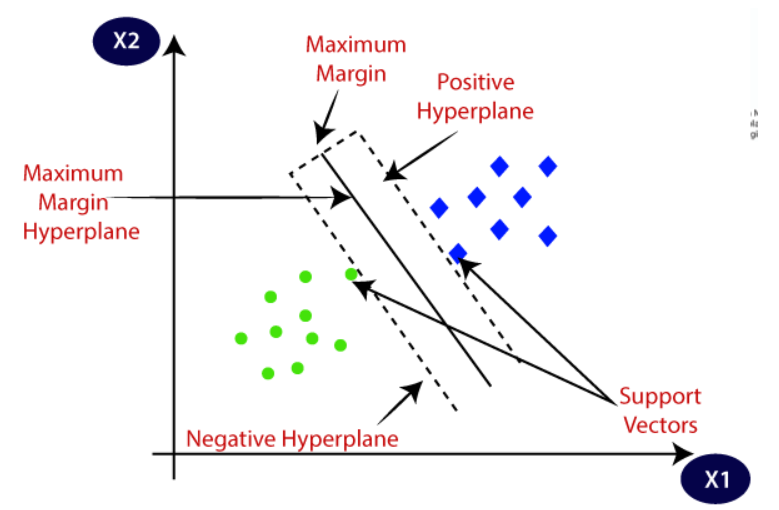


Figure 6 SVM Algorithm (Javatpoint, 2021)

Logistic regression (LR) uses thresholding to predict values of continuous variables. This constitutes in a better fit to solving linear problems using the best fitting line for thresholding which is the basis for binary classification. Because of its ability to predict values, LR algortihms are suitable for implementation when data is missing or unknown compared to other algorithms. This ability can also be paired with other supporting functions to aid prediction in continuous and categorical variables (Arabameri et al., 2018). LR is comparatively easier to implement, with its simplicity making it a favourable machine learning algorithm in the market. However, the downsides to this show in its inability to support and solve non-linear problems due to the nature of binary classification and the assumptions it makes for a linear approach. Similarly, another major limitation is the tendency of overfitting when the model becomes complex, which in accordance with the sensitivity means that accuracy takes a significant dip (Almoallem et al., 2021).



Figure 7 Logistic Regression (Javatpoint, 2021)

Diagram

Description automatically generated

Figure 8 Architecture Framework Flowchart

# Part 2

## Q1. Software to implement Artificial Intelligence

The software used in our project is Jupyter notebook, for which the code is submitted alongside this document. In addition, the code is shown in the Appendix section.

## Q2. Plotting Experimental Data

Indicators used: visited resources, announcement views, discussion, student absence days

### K-Nearest Neighbors (KNN) Model

Visit Resources

A picture containing rectangle

Description automatically generated

Figure 9 Visit resources (KNN)

Announcement Views

A picture containing rectangle

Description automatically generated

Figure 10 Announcement Views (KNN)

Discussions

Graphical user interface

Description automatically generated with low confidence

Figure 11 Discussions (KNN)

Absence

A picture containing table

Description automatically generated

Figure 12 Absence (KNN)

### Decision Tree Model

Visit resources

A picture containing shape

Description automatically generated

Figure 13 Visit Resources (Decision Tree)

Announcement Views

A picture containing graphical user interface

Description automatically generated

Figure 14 Announcement Views (Decision Tree)

Discussion

Graphical user interface

Description automatically generated with low confidence

Figure 15 Discussions (Decision Tree)

Absence

A picture containing text

Description automatically generated

Figure 16 Absence (Decision Tree)

### Support Vector Machine (SVM) Model

Visit resources

A picture containing shape

Description automatically generated

Figure 17 Visit Resources (SVM)

Announcement Views

A picture containing graphical user interface

Description automatically generated

Figure 18 Announcement Views (SVM)

Discussion

Graphical user interface

Description automatically generated with medium confidence

Figure 19 Discussions (SVM)

Absence

A picture containing text

Description automatically generated

Figure 20 Absence (SVM)

### Logistic Regression Model

Visit Resources

A picture containing shape

Description automatically generated

Figure 21 Visit Resources (Logistic Regression)

Announcement Views

A picture containing graphical user interface

Description automatically generated

Figure 22 Announcement Views (Logistic Regression)

Discussion

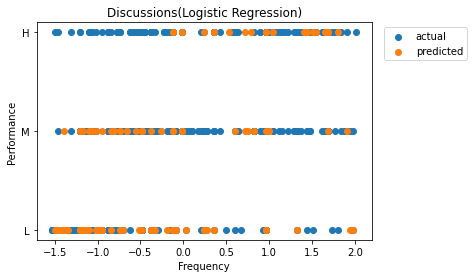


Figure 23 Discussions (Logistic Regression)

Absence

A picture containing text

Description automatically generated

Figure 24 Absence (Logistic Regression)

## Q3. Software features, results, and critical explanations

### Import libraries to training

Graphical user interface, text

Description automatically generated

Figure 25 Library imports for training

As we aim to train the models for use later on, for use of software and related techniques, the initial standard imports for libraries are pandas for creation of dataframes, numpy for mathematical operations and matplotlib’s pyplot used for the creation, labelling and plotting of charts.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 26 Dataset head

A picture containing graphical user interface

Description automatically generated

Figure 27 Selected dataset

The obtained dataset is then imported into Jupyter Notebook. We confirm the successful loading of the import by printing the head of the dataset. We then create a subset of the dataset encompassing five columns to house the subset. The subset data contains all selected data for our approach which will be used as the test and training data, namely ‘VisiTed Resources’, ‘

Text

Description automatically generated

Figure 28 Split the train and test dataset

AnnouncementsView’, ‘Discussion’, ‘StudentAbscenceDays’ and ‘Class’ respectively. We again print the head of the dataset to ensure it has been successfully created.

Graphical user interface, text

Description automatically generated

Figure 29 Get y dependent variables

Text

Description automatically generated

Figure 30 Classify absence

The dataset is then split into separate training and testing sample sets for the models. To do this, we import train\_test\_split from the sklearn.model\_selection library. The test and training sample sets are then assigned with a 20% skew for the testing sample size. The seed is set to 4 for the random generator. We then view the training and test sets along with output for the first five sets. Class values here can be assigned a value of ‘L’ for low, ‘M’ for medium and ‘H’ for high, and are used for the prediction of student performance based on other student factors within the data subset. Additionally, we converted ‘StudentAbscenceDays’ to an effectively true/false format of 0/1 for integration into model training as the default values from the original dataset were not able to be processed.

### Model Implementation

After preparing the training and testing dataset, we move into implementing 4 different machine learning models to demonstrate the prediction of student performance based on the 4 chosen indicators. The first algorithm we have chosen is the K-Nearest Neighbors (KNN) algorithm. We first train the model and predicted the result. We then used the accuracy score metric function to obtain the highest accuracy and the number of k neighbours that achieve the highest accuracy. We then rebuild the model with the best k value. The other 4 algorithms we used are decision tree classifier, support vector machine, and logistic regression for comparison to conclude which algorithm is the best in classifying this dataset.

Graphical user interface, text, application

Description automatically generated

Figure 31 Pre-processing for X variables

Graphical user interface, text, application

Description automatically generated

Figure 32 Finding k value with best accuracy

Graphical user interface, text, application, email

Description automatically generated

Figure 33 Training other models

The following diagrams show the methods we have used to evaluate the performance of the 4 algorithms. We have used the Jaccard index, F1-score and accuracy score to evaluate the models.

Text

Description automatically generated

Figure 34 Testing and Analysis

After performing the training of the models and evaluation, we plot the result of the predicted values against the indicator frequency. An example of the Visit Resources indicator is shown in the figure below. We first extract the indicator values from the training dataset and plot them against the training performance values on the graph. We also plot the predicted values on the graph for comparison with the actual values. We have included a title for each of the plots in each indicator and model, together with the x label, y label, and a legend.

Text

Description automatically generated

Figure 35 Visit Resource plot

The below diagram shows the code and result for the predicted results of student performance against the absence days. As mentioned above, the absence days are classified into 2 groups, under 7 days of absence and over 7 days. As the absence of data only has 2 groups, we are unable to use scatter plots to visualize the results. We use the "groupby" function for dataframes to group the two absence types and visualize the ratio between the different classes of performance within each absence group. The -0.812957 value in the output represents under 7 days of absence and the 1.23007 value represents over 7 days of absence.

Text

Description automatically generated

Figure 36 Absence days visualization

### Result Explanation

#### Visit Resource

In the prediction results for Visit Resource, the predicted results mostly matches with the actual data, which had showed a correlation between frequency of visiting resources, and academic performance. Identically, the prediction matched what the actual data classified. To be specific, higher Resource Visits correlate with high academic performance. However, due to the great variance in frequency of visit resources of M classified students in the test data, the prediction model classified M to many students with largely varying frequencies of resource visits. All models shared similar results, with Decision Tree having a slightly more sparse H classifications.

#### Announcement Views

The general prediction followed a similar pattern to Visit Resources, where lower frequency of announcement views would result in lower classification, and higher frequency of announcement views would result in higher classification, which mostly lines up with the actual data. But in contrast to the Visit Resource indicator, the prediction label for M class ended up mainly with students with low frequency in Announcement Views. This pattern correctly ties with the actual data. In terms of the individual models performance, KNN had slightly better accuracy overall in correlation to the actual data, but generally all perform similarly.

#### Discussions

When it came to the training data for Discussions, the labelling for each classification had a wide spread of range over the frequency of Discussion Posts. Consequently, it is difficult to point out a general pattern, and as such, most of the prediction models had difficulty matching the actual data, with the most accurate model being KNN, with an F1 score of 0.73, as opposed to the other models, all of which scored below 0.7.

#### Absence

Before training, the absence data was preprocessed and converted into 1s and 0s for each student, where 1 is absence days above 7, and 0 is absence days below 7. Since the values are normalized, the model outputs the absence data in a range of -2.0 to 2.0, but regardless of the value, if the value is below 0, it describes the student with low absence rate, and above 0 would be high absence rate. Subsequently, the general output of the models showed that students with high absence rates are generally predicted with Low and Medium academic performance, with 66-85% of the students predicted as Low academic performance. In contrast, about half of the students with low absence rates are classified with High academic performance. This correlates to the actual data, which showed that absence rate and academic classification is strongly linked. In all models, students with high absence rates are only classified with Low and Medium. However, for students with low absence rates, KNN, Decision tree, and Logistic Regression had predictions with all classifications, but SVM did not predict any students with Low academic performance.

### Model Evaluation

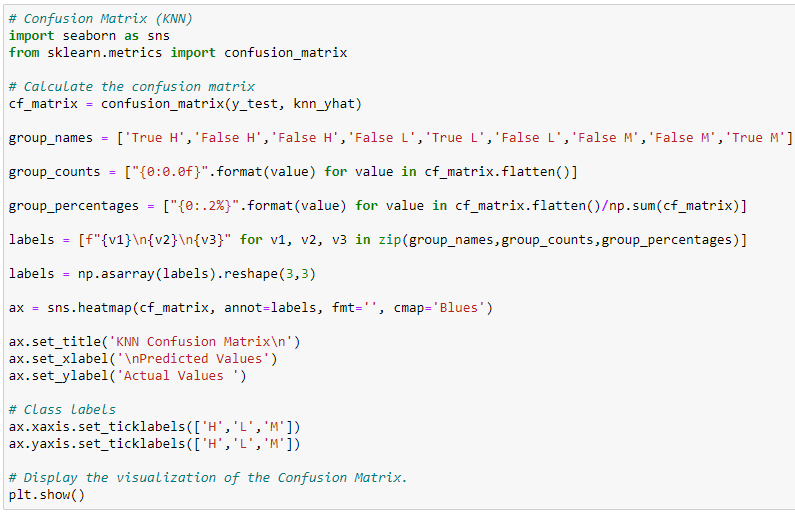


Figure 37 Plotting KNN Confusion Matrix

Chart, treemap chart

Description automatically generated with medium confidence

Figure 38 KNN Confusion Matrix

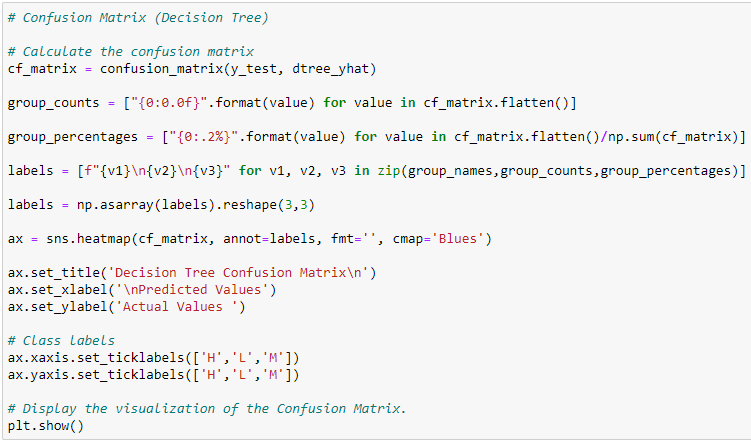


Figure 39 Plotting Decision Tree Confusion Matrix

Chart, treemap chart

Description automatically generated

Figure 40 Decision Tree Confusion Matrix

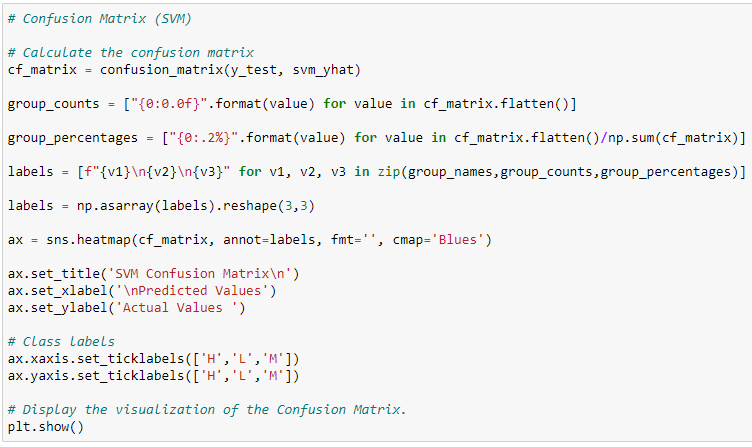


Figure 41 Plotting SVM Confusion Matrix

Chart, treemap chart

Description automatically generated with medium confidence

Figure 42 SVM Confusion Matrix

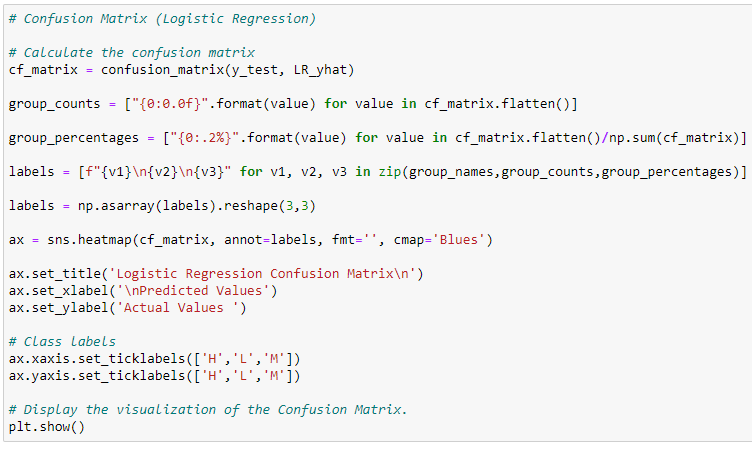


Figure 43 Plotting Logistic Regression Confusion Matrix

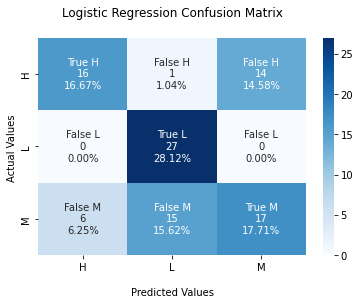


Figure 44 Logistic Regression Confusion Matrix

The confusion matrix shows the performance of the classification algorithm by plotting the percentage of predictions based on the color shading. The correct predictions are labelled True H, M, and L respectively. If the color shadings are darker on these squares compared to the False squares, it illustrates that the performance is better. In this case, KNN has the best results out of all the models.

Text

Description automatically generated

Figure 45 Plotting F1 scores of each Model

Before plotting the model’s performance, the seaborn library is imported to visualize the statistic data. This research used four algorithms, including KNN, Decision Tree, SVM, and Logistic Regression. The diagram above shows the technique used to plot a bar chart for the F1 Scores of the mentioned algorithms. First, all the F1 Scores are calculated and stored in a list ‘algo\_f1, ’ while the ‘algo’ list stores each algorithm’s name. After that, all the parameter is converted into a data frame as the seaborn only accepts the data frame to plot a bar chart. Then, each bar in the chart is given an annotation to show the number of F1 Scores in two decimals. In the end, the bar chart is given a title called “F1-Scores Comparison.”

Chart, bar chart

Description automatically generated

Figure 46 Model F1-Scores Chart

The bar chart above indicates the F1 Scores of each algorithm. F1 Score is a method combining **precision** and **recall** into a metric for the purpose of comparing the algorithm’s performance. The F1 Score ranges from 0 to 1. The higher the score, the better the algorithm’s performance. Among the KNN, Decision Tree, SVM, and Logistic Regression, KNN gained the best F1 Score with 0.73, while Logistic Regression had the lowest score with 0.61. Hence, KNN will be a better choice for developing the proposed model.

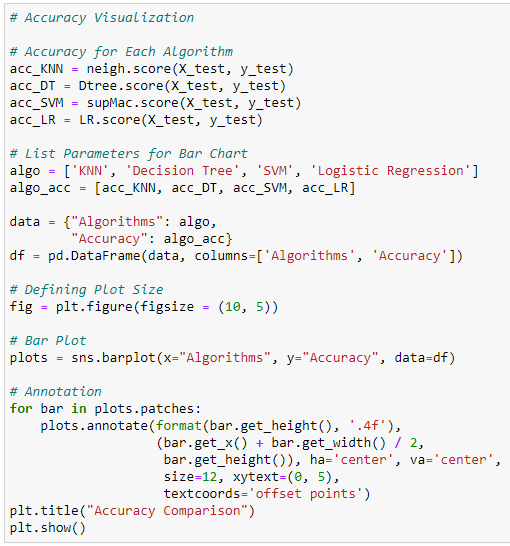


Figure 47 Visualizing Model Accuracy

Similar to the F1 Score’s bar chart, the diagram above shows the technique used to plot a bar chart for each algorithm’s accuracy. The only difference is F1 Score was changed to Accuracy.

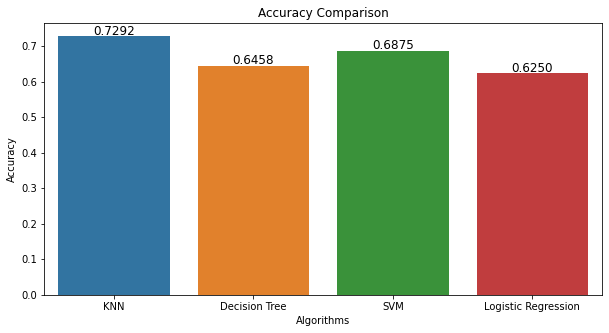


Figure 48 Model Accuracy Chart

The bar chart above indicates the accuracy of each algorithm. Accuracy is the most common approach used in machine learning to represent the algorithm’s performance as it is easy to understand and carry out. The accuracy ranges from 0% to 100%. The higher the percentage, the better the algorithm’s performance. Having a high accuracy means that the prediction made by the model is trustworthy and dependable while making decisions. Among the KNN, Decision Tree, SVM, and Logistic Regression, KNN gained the best accuracy with 72.92%, while Logistic Regression had the lowest accuracy with 62.50%. Combining the F1 Scores result, KNN is the best algorithm with the highest accuracy and F1 Score to develop the proposed model.

# References

Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. B. S. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. In *Big Data* (Vol. 7, Issue 4, pp. 221–248). Mary Ann Liebert Inc. https://doi.org/10.1089/big.2018.0175

Aljarah, I. (2016) *Students' academic performance dataset*. Kaggle. Retrieved August 21, 2022, from https://www.kaggle.com/datasets/aljarah/xAPI-Edu-Data

Almoallem, Y. D., Taha, I. B. M., Mosaad, M. I., Nahma, L., & Abu-Siada, A. (2021). Application of logistic regression algorithm in the interpretation of dissolved gas analysis for power transformers. *Electronics (Switzerland)*, *10*(10). https://doi.org/10.3390/electronics10101206

Althubaiti, A., Bamuhair, S., Al Farhan, A., ur Rahman, S., & Al-Kadri, H. (2016). Class attendance and cardiology examination performance: a study in problem-based medical curriculum. *International Journal Of General Medicine*, 1. https://doi.org/10.2147/ijgm.s96627

Arabameri, A., Pradhan, B., Rezaei, K., Yamani, M., Pourghasemi, H. R., & Lombardo, L. (2018). Spatial modelling of gully erosion using evidential belief function, logistic regression, and a new ensemble of evidential belief function–logistic regression algorithm. *Land Degradation and Development*, *29*(11), 4035–4049. https://doi.org/10.1002/ldr.3151

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2019.10.118

Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20–28. https://doi.org/10.38094/jastt20165

Fadelelmoula, T. (2018). The impact of class attendance on student performance. *International Research Journal Of Medicine And Medical Sciences*, *6*(2), 47-49. https://doi.org/10.30918/irjmms.62.18.021

Ferrari, M. (2019). *Data Analysis - student's behavior*. Kaggle. Retrieved August 21, 2022, from https://www.kaggle.com/code/marlonferrari/data-analysis-student-s-behavior/notebook

Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., & Yang, H. (2019). A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications*, *115*, 356–372. https://doi.org/10.1016/j.eswa.2018.08.021

Javatpoint. (2021). *K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint*. www.javatpoint.com. Retrieved 25 August 2022, from https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning.

Javatpoint. (2021). *Logistic Regression in Machine Learning - Javatpoint*. www.javatpoint.com. Retrieved 25 August 2022, from https://www.javatpoint.com/logistic-regression-in-machine-learning.

Javatpoint. (2021). *Machine Learning Decision Tree Classification Algorithm - Javatpoint*. www.javatpoint.com. Retrieved 25 August 2022, from https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

Javatpoint. (2021). *Support Vector Machine (SVM) Algorithm - Javatpoint*. www.javatpoint.com. Retrieved 25 August 2022, from https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm.

Kuzminykh, I., Ghita, B., & Xiao, H. (2021). The relationship between student engagement and academic performance in online education. *2021 5th International Conference on E-Society, E-Education and E-Technology*. https://doi.org/10.1145/3485768.3485796

Pant, R., & Trafalis, T. B. (2015). Svm classification of uncertain data using robust multi-kernel methods. *Springer Proceedings in Mathematics and Statistics*, *130*, 261–273. https://doi.org/10.1007/978-3-319-18567-5\_13

Patel, H. H., & Prajapati, P. (2018). Study and Analysis of Decision Tree Based Classification Algorithms. *International Journal of Computer Sciences and Engineering*, *6*(10), 74–78. https://doi.org/10.26438/ijcse/v6i10.7478

Pisner, D. A., & Schnyer, D. M. (2019). Support vector machine. In *Machine Learning: Methods and Applications to Brain Disorders* (pp. 101–121). Elsevier. https://doi.org/10.1016/B978-0-12-815739-8.00006-7

Zhang, W., Zhou, Y., & Yi, B. (2019). An interpretable online learner’s performance prediction model based on Learning Analytics. *Proceedings of the 2019 11th International Conference on Education Technology and Computers*. https://doi.org/10.1145/3369255.3369277