**UNIVERSITI TUNKU ABDUL RAHMAN**

**FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY**

**A blue background with white text and red letters

Description automatically generated**

**Session: June 2024**

**UCCB3224 DATA MINING TECHNIQUES**

**Group 7**

**Member List:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Group Member Name** | **Student ID** | **Programme** | **Signature** |
| 1 | Ng Yi Jian | 22ACB07029 | IA | **A black scissors and a white background  Description automatically generated** |
| 2 | Lee Zong Hao | 22ACB06561 | IA |  |
| 3 | Chin Wai Teng | 22ACB00233 | IB | A black background with a black square  Description automatically generated with medium confidence |
| 4 | Teh Shu Hui | 22ACB00182 | IB | A close-up of a sign  Description automatically generated |
| 5 | Pua Ming Sien | 19ACB06537 | IB | A signature of a person  Description automatically generated |

**Assignment Marksheet**

By signing below, we confirmed that the work produced was original and purely based on our own sentence construction. Should there be any plagiarism detected, we agreed mark penalization on the part(s) detected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Name** | **ID** | **Practical Group** | **Program (IA/IB/DE)** | **Signature** |
| 1 | Ng Yi Jian | 2207029 | P3 | IA | **A black scissors and a white background  Description automatically generated** |
| 2 | Lee Zong Hao | 2206561 | P3 | IA |  |
| 3 | Chin Wai Teng | 2200233 | P1 | IB | A black background with a black square  Description automatically generated with medium confidence |
| 4 | Teh Shu Hui | 2200182 | P1 | IB | A close-up of a sign  Description automatically generated |
| 5 | Pua Ming Sien | 1906537 | P1 | IB | A signature of a person  Description automatically generated |

| **Criteria** | **Weightage** | **Member 1** | **Member 2** | **Member 3** | **Member 4** | **Member 5** |
| --- | --- | --- | --- | --- | --- | --- |
| **Individual:** | **Name** | Ng Yi Jian | Lee Zong Hao | Chin Wai Teng | Teh Shu Hui | Pua Ming Sien |
| Final   Deliverables (Quality of Work) | 4%  (Scale\*4 / 5) |  |  |  |  |  |
| **Team:** |  |  |  |  |  |  |
| Report and Documentation | 3%  (Scale\*3 / 5) |  |  |  |  |  |
| Effort and Technical Capability (Overall as a Team) | 3%  (Scale\*3 / 5) |  |  |  |  |  |
| Final Deliverables (Quality of System) | 3%  (Scale\*3 / 5) |  |  |  |  |  |
| Innovativeness | 2%  (Scale\*2 / 5) |  |  |  |  |  |
|  | **15%** | % | % | % | % | % |

**Marking Scheme**

|  |  |
| --- | --- |
| **Scale (0-5)** | **Description** |
| 5 | Excellent work produced. Evidence of in-depth study **and** critical thought |
| 4 | Good work produced. Evidence of in-depth study **or** critical thought |
| 3 | Average work produced. Evidence of adequate study **or** thought/idea, although not extensively covered |
| 2 | Below average work produced. Evidence of some study **or** thought/idea, but not quite adequate. |
| 1 | Poor work performance **or** work **not** supported by any study/basis |
| 0 | Not attempted |

**Executive Summary(Contribution by Ng Yi Jian)**

The purpose of the study is to investigate how big data and artificial intelligence (AI) may be integrated into Massive Open Online Courses (MOOCs) to overcome the difficulties associated with evaluating student performance and offering prompt assistance. Because MOOCs provide worldwide self-directed learning and open-access resources, they have completely changed the educational landscape. They have difficulties, though, in correctly assessing student performance and helping students who are having difficulty.

The study focuses on using a dataset from a Learning Management System (LMS), which provides specific information on how students engage with the MOOCs platform, to address these problems. This dataset offers a thorough understanding of student involvement and performance by including their actions during educational videos as well as their responses to exercise questions.

This has been suggested that big data analytics and artificial intelligence be used to transform this unprocessed educational data into useful insights. The goal is to improve teaching and learning for instructors and students by creating predictive models. This strategy aims to enhance performance evaluations, customize assistance to meet specific requirements, and eventually enhance the total learning experience provided by MOOCs.

Table of Contents

Executive Summary1

Table of Contents2

1.0 Business Understranding3

Introduction & Background Information 3

Problem Statement 4

Objectives 5

Motivation 5

Current Existing Solution 6

Performance Measurement 6

Data Mining Success Criteria 7

Project Plan (Gantt Chart) 7

Development Tools 8

2.0 Data Understanding11

3.0 Data Preparation15

4.0 Modeling18

Feature Selection 20

Hyperparameter Tuning 20

5.0 Evaluation24

Evaluation Measures 24

Results 25

6.0 Deployment27

Integration into MOOC Platform 27

Monitoring and Maintenance Plan 27

7.0 Conclusion28

References29

**1.0 Business Understanding (Contribution by Chin Wai Teng)**

**Introduction & Background Information**

Massive open online courses (MOOCs) are primarily free web-based distance learning programs meant for huge groups of geographically dispersed students. MOOCs resemble college courses but are loosely structured. While these courses do not usually award academic credit, they frequently provide credentials, better job chances, or opportunities for additional study. MOOCs are commonly utilized for higher education, skill building, and career growth [1].

Next, MOOCs work as online learning courses that students can access via the Internet. Typically, these courses are delivered via cloud computing platforms. The course content is developed using course authoring tools and then hosted on a learning management systems (LMS) platform. Course materials and teachers are provided by the course provider, which is typically a university. The LMS platform, such as EdX, Canvas, Coursera, or Udacity, provides the technology foundation for course modules, user access, and other learning resources [1].

Dave Cormier of the University of Prince Edward Island in Canada invented the term “MOOC” (pronounced “kook”) in 2008 which describes an online course provided by the University of Manitoba. The “Connectivism and Connective Knowledge” course was taken online by 25 tuition-paying university students and 2300 non-paying general public students. There were RSS feeds for the information, and participation was made possible through a variety of channels, including the Moodle LMS, blog entries, the Second Life online virtual environment, and real-time online meetings. In 2011, the Massachusetts Institute of Technology (MIT) launched OpenCourseWare, the first substantial collection of MOOC resources made available by a university. In 2012, MIT and Harvard University launched the EdX effort to promote MOOCs [1].

As the demand for technological jobs grows, so will the popularity of online courses. As a result of the COVID-19 epidemic, MOOCs are emerging as the new norm for education and specialization at all levels, from elementary school to bachelor’s and master’s degree programs, as part of remote learning plans. Enrolment in MOOCs increased dramatically during the pandemic, and numerous additional MOOCs have subsequently been introduced [1]. Despite this, the quality of assessment in MOOCs remains inconsistent. Assessment practices in MOOCs and other educational technologies sometimes fall short of the current state of the art [3].

**Problem Statement**

The educational scene has changed dramatically in recent years, thanks in large part to technological improvements. Massive Open Online Courses (MOOCs) have emerged as a significant development in the field of online education. MOOCs have become extremely popular as an alternate and accessible way of learning but it still remains a huge problem to effectively measure student performance and identify those who may require more assistance. Traditional evaluation approaches may fail to reflect the intricacies of online learning behavior, creating a gap in timely and effective responses.

Common issues in evaluating MOOCs include lack of established criteria, low completion rates, varied instructor involvement and accessibility. MOOCs are a relatively new phenomenon, with no standardized assessment criteria. These courses vary widely and do not follow a standardized, agreed definition or format. MOOCs do not often have predefined learning objectives that apply to all participants. Individuals might create their own goals and objectives, or none at all. These courses are challenging to evaluate since they lack validated assessment criteria and include a wide range of learning objectives. Traditional course assessment approaches, which are based on classroom instruction, do not easily apply to MOOCs. Next is the lack of completion rate, as thousands of people join MOOCs, yet only a small percentage complete their courses.Completion rates are an unreliable metric of MOOC performance. These rates do not account for the range of participant learning experiences. Some students get the information that they need from a specific subject but do not complete the course. Other people passively study or merely participate in conversations, but again they do not finish the course [2].

Another problem with evaluating MOOCs is that the instructor’s role fluctuates. In some courses, the instructor is only a facilitator, and in others, the teacher is the primary subject matter expert. The vast number of participants complicates the development of an instructor-student relationship. The large number of students in MOOCs renders individual student attention unfeasible. As a result, students may feel detached and disengaged with the course. The function of the instructor, combined with the wide range of student experiences, makes assessing teaching efficiency extremely challenging. Last but not least, one of the advantages of MOOCs is that they are available online to anybody. However, this raises the question of accessibility. As certain MOOCs rely largely on multimedia content, accessibility must be considered. Everyone should be able to access videos, presentations, audio courses, social media debates, and other content. Videos, for example, should have captions so that all students can interact with the content. Accessibility and varying student information literacy skills make evaluating MOOCs challenging. If participants are unable to access, comprehend, or engage with course information, assessing learning will be challenging [2].

**Objectives**

1. To precisely predict students’ performance based on input features.
2. To implement data exploration (EDA) and preprocessing techniques to prepare data for machine learning.
3. To apply various machine learning algorithms to predict students’ performance.
4. To evaluate and analyze student performance data.
5. To optimize the selected machine learning algorithm to enhance predictive accuracy.

**Motivation**

The motivation for this project stems from the potential to improve the educational experience in MOOCs through fast and accurate assessments of student performance. The initiative aims to solve existing gaps in student support within MOOCs using machine learning algorithms, resulting in enhanced learning outcomes and more personalized educational experiences.

**Current Existing Solution**

A look at evaluation of assessment systems in MOOCs ( Massive Open Online Courses), starting with Stanford AI Courses in 2011. Initially, MOOCs like the Stanford course used basic assessment kinds like multiple-choice and numerical responses, but they ingeniously blended these tests within instructional videos to provide ongoing formative feedback. Coursera and MITx (now Open EdX) followed suit with their platforms, providing features such as problem banks for mastery learning, programmatically scored assignments, and increasingly complicated evaluations that included circuit simulations and algorithm outputs [3].

Despite advances in assessment technology, the bulk of MOOCs still use simple evaluation methods like multiple-choice questions and fill-in-the-blank exercises. While systems such as EdX provide a variety of assessment methods, only a small proportion of courses properly utilize them. These methods are similar to the more sophisticated, integrated evaluations seen in computer-based learning environments such as Cognitive Tutor and ALEKS, which tailor tests to the learner’s progress. The problem is to scale these sophisticated evaluation methodologies to the diversified and large-scale nature of MOOCs, which is compounded by the fact that many MOOC teachers are subject matter experts with no experience in educational technology [3].

**Performance Measurement**

Performance measurement in this project will be comprehensive, with an emphasis on both the accuracy and practical usability of the prediction models. Key performance indicators will include conventional machine learning measures like accuracy precision, recall, F1-score, and area under the ROC curve (AUC-ROC), which will assess the model’s capacity to correctly classify student performance and identify at-risk students. Furthermore, the timeliness of interventions will be evaluated by measuring the time it takes for the system to recognize challenging students and the effectiveness of these interventions in improving student outcomes.

Student engagement and learning outcomes will also be important KPIs, with measures such as the amount of time students spend on the platform, the number of interactions with course content, completion rates and learning gains as measured by scores.

**Data Mining Success Criteria**

One of the data mining success criteria is accurate prediction of student performance. The effectiveness of the data mining process will be determined by the machine learning algorithms’ ability to predict student performance in MOOCs. The algorithms should provide accurate predictions for grades, completion rates, and other performance metrics with low error rates. Next, balanced algorithm performance. Algorithms must be built to prevent overfitting and underfitting. Success will be evaluated by the algorithm’s capacity to generalize to new data, ensuring consistent performance across both training and testing dataset. This balance will be important to the algorithm’s actual use.

Timely detection of at-risk students. The algorithms should effectively identify students at-risk of poor performance or dropping out early enough to provide appropriate help. Moreover, discovery of significant patterns and insights in the dataset. Patterns should provide valuable insights into student behavior, performance, and engagement in MOOCs.

**Project Plan- Gantt Chart**

**A screenshot of a calendar

Description automatically generated**

**Figure 1 Timeline**

**Development Tools**

Several development tools have been used by our team, one of the development tools is Google Colab. Google Colaboratory, or Colab, is an as-a-service version of Jupyter Notebook that allows users to create and run Python code in the browser. Google Colab is based on Project Jupyter source and supports Jupyter notebooks without requiring any local software installation. However, unlike Jupyter notebooks, which support many languages such as Python, Julia, and R, Colab presently only supports Python. Colab notebooks are kept on a Google Drive account and can be shared with other users, just like any other Google Drive file. The notebooks also have an autosave feature, but they do not permit simultaneous editing, therefore collaboration has to be serial rather than parallel [4]. The reason that our team uses Google Colab is due to Colab is a highly accessible and user-friendly platform that requires no setup, and allows users to start coding right away in the web browser. It is a cloud-based environment that eliminates the need for strong local hardware by providing free access to GPU and TPUs, which can dramatically expedite the training of machine learning models, particularly deep neural networks. Furthermore, Colab’s seamless interface with Google Drive allows for simple access to data and files, while its real-time collaboration features allow team members to work together efficiently by sharing and modifying notebooks concurrently.

Second development tool that has been used by our team is Python. Python is a high-level, interpreted, object-oriented programming language with dynamic semantics. Due to its reputation as a beginner-friendly language, Python has displaced Java as the most popular introductory language. This is because it takes care of a lot of the complexity for the user, freeing them up to concentrate on understanding programming concepts rather than specifics [5]. Selecting Python as one of our development tools is due to its adaptability, user-friendliness, and robust ecosystem. A comprehensive toolkit covering every stage of the data ming process, from data cleaning and exploration to model building and evaluation, is provided by Python’s vast library ecosystem, which includes Pandas for data manipulation, Numpy for numerical operations, and Scikit-learn for machine learning.

Third development tool that has been used by our team is Python Matplotlib. Matplotlib is a cross-platform data visualization and graphical plotting library which includes histograms, scatter plots, and bar charts for Python and its numerical extension Numpy. As such, it represents a potential open-source alternative to MATLAB. Developers can also utilize matplotlib’s APIs (Application Programming Interfaces) to include charts in GUI apps. The structure of a Python matplotlib script makes it possible to create a visual data plot with just a few lines of code in most cases [6]. The reason that our team applied matplotlib is because matplotlib is a fundamental data visualization tool in the Python environment and a must-have for any data mining project. With its high degree of customization, it can accurately build a variety of static, animated, and interactive plots.

Fourth development tool that has been used by our team is Scikit-learn (Python).  Within the Python community, Scikit-learn is the go-to library for Machine Learning (ML) and is available as an open source library. With the use of Machine Learning (ML), computers may create and train prediction models without the need for explicit programming by learning from incoming data [7]. The reason that our team uses Scikit-learn is due to its vast flexibility, speed, and ease of use make it an invaluable tool for Python data mining and Machine Learning. It is appropriate for a variety of predictive modeling jobs since it offers a strong selection of well-optimized algorithms for dimensionality reduction, clustering, regression, and classification. Moreover, easy experimentation and quick model construction are made possible by the library’s clear and consistent API design, which covers everything from data pretreatment to model evaluation and Scikit-learn easily interfaces with other Python libraries, such as Matplotlib, Numpy, and Pandas, facilitating a fluid workflow from data manipulation to visualization.

Next is NumPy which is a core Python library for scientific computing. A multidimensional array object, different derived objects like masked arrays and matrices, and a variety of routines for quick array operations like sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation [8]. The reason for selecting it is large, multi-dimensional arrays and matrices are supported by NumPy, the core library for numerical computing in Python, which also offers a range of mathematical functions to manipulate the arrays. Large datasets and intricate mathematical calculations, which are typical in data mining, can be handled effectively with NumPy and compared to conventional Python lists, NumPy’s vectorized operations enable quicker and more memory-efficient computations. Moreover, NumPy is the foundation of several other data science libraries, including Pandas and Scikit-learn which make it an essential part of the data mining ecosystem.

Last but not least, Pandas also is one of the development tools that have been used by our team. Pandas is used to work with data collections. Its features include data manipulation, cleansing, analysis, and exploration [9]. The reasons why our team uses Pandas are because it is a powerful data manipulation and analysis package based on NumPy and optimized for structured data. It has two core data structures, Series and DataFrame, which allow for easy labeled data processing, making it perfect for working with tabular datasets that are popular in data mining. Pandas make data cleaning, filtering, aggregating, and transformation processes easier, allowing users to swiftly prepare the data for analysis and modeling. Pandas’ interaction with other Python libraries such as Matplotlib for visualization and Scikit-learn for machine learning expands its utility, making it a must-have tool in any data mining workflow.

**2.0 Data Understanding (Contribution by Pua Ming Sien)**

In data understanding, it involves a detailed look at a dataset containing various student attributes, such as demographics, academic activities, and performance. The goal is to identify important features, perform exploratory data analysis (EDA), and establish insights that support business objectives.

The dataset includes 17 attributes: gender, nationality, place of birth, stage ID, grade ID, section ID, topic, semester, relation, raised hands, visited resources, announcements view, discussion, parent answering survey, parent school satisfaction, student absence days, and class. It contains 480 entries, with all columns having non-null values, which means there are no missing data in this dataset. The dataset has a mix of data types: 4 numeric columns (raised hands, visited resources, announcements view, discussion) and 13 categorical columns (e.g., gender, nationality, place of birth, and others.). The target field in this dataset is "Class," which categorizes student performance into three levels. Include low (0-69), middle (70-89), and high (90-100).

The EDA process starts with visualizing key relationships in the data. First is a bar chart. Figure 2.1 presents the distribution of students across different performance levels (low-level, middle-level, high-level), which helps to understand the overall performance distribution.

A graph of a student performance classification

Description automatically generated

**Figure 2.1 Distribution of students performance classification**

Next,  Figure 2.2 uses a scatter plot to explore the relationship between raisedhands and VisITedResources, helping to identify the relationship between the number of times students raised their hands in class and the number of resources they visited. The positive correlation shown suggests that students who actively participate in class (raise hands) also tend to use  more resources.

A chart of different colored dots

Description automatically generated

**Figure 2.2 Scatter plot for raised hands and visited resources**

Furthermore, Figure 2.3 is a bar chart that illustrates the average level of parent satisfaction by different educational stages (lower level, middle school, high school). It shows that parents of lower-level students report higher satisfaction compared to those in middle and high school. This could indicate that parents perceive a decline in satisfaction as students progress through grades.

A graph of different colored squares

Description automatically generated

**Figure 2.3 Average level of parent satisfaction by different educational stages**

Figure 2.4 presents a box plot comparing the number  of discussion participations between male and female students. This may show whether one gender participates more actively in discussions, showcasing potential disparities in engagement.

A blue and orange rectangular boxes

Description automatically generated

**Figure 2.4 Discussion participation by gender**

Additionally, Figure 2.5 is a histogram showing the distribution of how many announcements students view. The data shows a right-skewed distribution, indicating that while most students check announcements a few times, some check them significantly more often, which could point to varying levels of engagement with school communications.

A graph with blue lines and a line

Description automatically generated with medium confidence

**Figure 2.5 Distribution of announcements viewed**

The correlation of "Class" with itself is 1.000, which is expected since any variable will always be perfectly correlated with itself. There is a weak negative correlation between "Class" and "Discussion," indicating that as students participate more in discussions, their class level may decrease slightly, though this relationship is weak. "AnnouncementsView" also shows a weak negative correlation, suggesting that students who view more announcements might be in slightly lower classes, but the connection is not strong. The weak negative correlation with "ParentschoolSatisfaction" implies that as parent satisfaction increases, students might be somewhat more likely to be in a lower class, although the relationship is not significant. A slightly stronger negative correlation is seen with "VisITedResources," meaning students who visit more resources tend to be in lower classes, but the relationship remains weak. "raisedhands" has the strongest negative correlation with "Class" among the attributes listed, suggesting that students who frequently raise their hands are more likely to be in lower classes, though this relationship is still not very strong. Overall, these analyses suggest that a student's involvement in different activities might be somewhat related to their class level. However, these relationships are not very strong, which means can't rely on these features alone to predict a student's class level.

In the analysis, several transformations were applied to improve data quality and consistency. To handle skewness, a log transformation was applied to the raisedhands feature, while a square root transformation was used for the VisITedResources feature. These transformations helped in making the distributions more symmetric. Additionally, features were normalized and standardized to maintain consistent scales. Categorical variables were encoded using label encoding for the Class feature and one-hot encoding for gender and StageID. To enhance the dataset, new attributes were created by combining existing features. Outliers were removed using the IQR method to clean the data further. Newly created attributes such as Engagement\_Ratio, Resource\_Ratio, and Discussion\_Ratio were calculated to explore additional relationships within the data. The updated correlation matrix, which includes these new attributes, helps in understanding their relationships with other features and the target variable. This comprehensive approach ensures a well-prepared dataset for further analysis and modeling.

**3.0 Data Preparation (Contribution by Ng Yi Jian)**

**A diagram of a data flow

Description automatically generated**

**A diagram of a process

Description automatically generated**

**A diagram of a diagram

Description automatically generated**

Finding any missing values in the dataset is the first step in the data preparation process. If a missing value is found, it needs to be deleted. Next, we must determine if the dataset contains duplicate values. In case a duplicate value is found, it will be removed from the dataset. The dataset needs to be divided into training and test sets. The training set and test set have a ratio of 0.8 to 0.2. The next step is to divide the features and target. The predictors are the columns that remain in X after we remove the target "Class." To make sure that y is a separate duplicate of the target dataset, we will select "Class" as the target in y.

Numerical and categorical features are handled differently in the define preprocessing pipeline step in order to get the data prepared for machine learning. A pipeline (num\_pipe) is created for numerical features, which scales features to have a mean of 0 and a standard deviation of 1 after first imputes missing values using the median and standardizing the data using StandardScaler. Another pipeline (cat\_pipe) is made specifically for categorical features; it imputes missing values using the most common category and then uses one-hot encoding to transform categorical values into a binary format. In combine pipeline part, a ColumnTransformer is created by combining these two pipelines, which then applies the relevant pipeline to the associated columns in the dataset.

During the fit and transform phase, the `fit\_transform()` method is utilized to apply the preprocessing pipeline to transform the features (X). This method fits the pipeline to the data by calculating statistics such as the median for imputation and then transforms the data correspondingly (e.g., scaling numerical features and one-hot encoding categorical ones). We may see the changes made to the dataset by viewing the whole preprocessing pipeline after applying the pipeline. `get\_feature\_names\_out()` may be used to get the feature names after transformation; this is particularly helpful after one-hot encoding as it displays the newly generated feature columns.

In label encoding part, we used to convert the labels for the target variable (y) from categorical to numerical format. First, LabelEncoder is fitted to the labels, and then the labels are applied to the converted numerical format. Lastly, print encoded classes may be used for analyzing the label classes. This helps decode the encoded labels by displaying the original categories that correlate to each numerical label.

**4.0 Modeling (Contribution by Lee Zong Hao)**

In this modeling phase, several basic machine learning algorithms have been chosen, which consist of Logistic Regression (LR), Decision Tree Classifier (DT), Random Forest Classifier (RF) and Support Vector Classification (SVC).We then fit the feature-only training set into the algorithms. After fitting all the models, K-fold validation is used where k is set to 10 to determine the accuracy of the model with 10 different folds. We compile all 10 accuracy for each model and calculate the mean to find out the average accuracy for each model. Result of the training mean accuracy score is shown in **Table 4.1.**

**A screenshot of a computer

Description automatically generatedFigure 4.1.1 Logistic Regression Mean Accuracy**

A screenshot of a computer

Description automatically generated**Figure 4.1.2 Decision Tree Classification Mean Accuracy**

**A screenshot of a computer

Description automatically generatedFigure 4.1.3 Random Forest Classification Mean Accuracy**

**A screenshot of a computer

Description automatically generatedFigure 4.1.4 Support Vector Classification Mean Accuracy**

|  |  |
| --- | --- |
| **Algorithms** | **Mean Accuracy Score (%)** |
| Logistic Regression | 75.93 |
| Decision Tree | 68.85 |
| Random Forest | 79.86 |
| Support Vector Classification | 75.16 |

**Table 4.1 Model Training Mean Accuracy Score**

After evaluating the training score for each model, we decided to choose the top 3 performers for further tuning which is the **LR**, **RF**, **SVC**.

**4.2 Feature Selection**

Variance Threshold has been used during feature selection. Variance Threshold removes features with low variance from the dataset. Features with low variance often contribute little to the model’s predictive power. For example, if a feature has almost similar value across all samples, it provides minimal information for distinguishing between classes. By removing features with low variance, we can reduce the number of features in our dataset. This reduction helps to simplify the model, improve computational efficiency and potentially reduce overfitting and data noises.

A screenshot of a computer

Description automatically generated

**Figure 4.2.1 Variance Threshold**

Figure 4.2.1, showed improvements in **Random Forest** models when implementing the Variance Threshold feature selection. However, Logistic regression and SVM still perform the same even after the feature selection compared to Table 4.1.

**4.3 Hyperparameter Tuning**

The models will further tune by using grid search to find the best parameter for each model for the prediction. However, to find out the best model, a pipeline which consists of feature selection and a pipeline with no feature selection are both made to be experimented.

**Logistic Regression**

Figure 4.3.1 below shows the parameters for GridSearchCV to evaluate.

**A computer screen shot of a code

Description automatically generated**

**Figure 4.3.1 : Parameter Grid for Logistic Regression**

After the GridSearch evaluation, the result is shown below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | | **Default Value** | **Best parameter Value** | |
| **With Threshold** | **Without Threshold** |
| **Parameter** | threshold |  | 0.01 |  |
| penalty | l2 | l2 | l2 |
| C | 1 | 1 | 1 |
| **Accuracy** | | 0.7593 | 0.7593 | 0.7593 |
| **STD** | | 0.0634 | 0.0602 | 0.0602 |

**Table 4.2: Performance Metric of Logistic Regression after hyperparameter tuning**

Table 4.2 shows that the Logistic Regression model’s accuracy remained unchanged despite the threshold adjustment, but the standard deviation slightly decreased with the threshold modification indicating a more consistent performance. The regularization parameters were not altered, reflecting that the threshold adjustment was sufficient to achieve this improvement in consistency.

**Random Forest**

Figure 4.3.2 below shows the parameters for GridSearchCV to evaluate.

**A computer code with numbers and letters

Description automatically generated**

**Figure 4.3.2: Parameter grid for Random Forest**

After the GridSearch evaluation, the result is shown below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | | **Default Value** | **Best parameter Value** | |
| **With Threshold** | **Without Threshold** |
| **Parameter** | threshold |  | 0.1 |  |
| n\_estimators | 100 | 100 | 100 |
| max\_depth | None | 30 | 10 |
| criterion | gini | entropy | gini |
| **Accuracy** | | 0.7986 | 0.8118 | 0.8037 |
| **STD** | | 0.0717 | 0.0482 | 0.0586 |

**Table 4.3 : Performance Metric of Random Forest after hyperparameter tuning**

Table 4.3 shows the threshold tuning has improved the model’s accuracy and lowering the standard deviation makes it more consistent. While the performance without threshold is slightly lower in both accuracy and consistency to the threshold-tuned model, it is still an improvement over the default settings.

**Support Vector Machine (SVM)**

Figure 4.3.3 below shows the parameters for GridSearchCV to evaluate.

**A computer code with numbers and symbols

Description automatically generated**

**Figure 4.3.3:Parameter grid for SVM**

After the GridSearch evaluation, the result is shown below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | | **Default Value** | **Best parameter Value** | |
| **With Threshold** | **Without Threshold** |
| **Parameter** | threshold |  | 0.01 |  |
| C | 1 | 0.1 | 1 |
| kernel | rbf | linear | linear |
| gamma | scale | scale | scale |
| **Accuracy** | | 0.7516 | 0.7725 | 0.7749 |
| **STD** | | 0.0675 | 0.0632 | 0.0499 |

**Table 4.4 : Performance Metric of SVM after hyperparameter tuning**

Table 4.4 shows the SVM model’s accuracy improved slightly with hyperparameter tuning in both with and without threshold adjustment. Notably, the accuracy increased from 0.7516 (default) to 0.7725 with the threshold and 0.7749 without it. However, the standard deviation decreased more significantly without the threshold modification, indicating a more consistent performance.

**5.0 Evaluation**

**5.1 Evaluation Measures**

The performance of the built model is important in the context of a project, so the results will be evaluated using evaluation measures such as accuracy, precision, recall and F1 score. Measures calculated using Table 5.1, which shows classification confusion matrix based on Equation 1,2,3 and 4 respectively.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Detected** | |
| **Positive** | **Negative** |
| **Actual** | **Positive** | True Positive (TP) | False Negative (FN) |
| **Negative** | False Positive (FP) | True Negative (TN) |

**Table 5.1 : Confusion Matrix**

Accuracy is the metric that measures the instance of accurate prediction from all of the available data. Precision is the ratio of correctly classified cases to the total number of misclassified cases and correctly classified cases. Recall is the ratio of correctly classified cases to the total number of unclassified cases and correctly classified cases. F1 score is the weighted harmonic mean of precision and recall which is considered a good indicator of the relationship between them.

**A black text on a white background

Description automatically generated**

**Equation 1: Accuracy**

**A black text on a white background

Description automatically generated**

**Equation 2: Precision**

**A mathematical equation with black text

Description automatically generated**

**Equation 3: Recall**

**A close-up of a sign

Description automatically generated**

**Equation 4: F1 Score**

**5.2 Results**

Evaluation is made on the test set after the hyperparameter tuning has been done.The best model chosen after hyperparameter tuning is Random Forest on both pipelines. Both Random Forest model with and without feature selection are both evaluated on the test set. The result metrics used are accuracy, recall, precision, f1-score, classification report, and confusion matrix to ensure the reliability of the model. Table 5.1 shows the performance of both sets on the test set.

|  |  |  |
| --- | --- | --- |
| **Model** | **RF**  **w/o  feature selection** | **RF**  **w feature selection** |
| **Criteria** |
| **Accuracy (%)** | 81.25 | 80.21 |
| **Recall (%)** | 81.25 | 80.21 |
| **Precision (%)** | 81.68 | 80.68 |
| **F1-score (%)** | 81.36 | 80.28 |

**Table 5.1 Evaluation results on test set**

|  |  |
| --- | --- |
| **A diagram of a confusion matrix  Description automatically generated** | **A diagram of a confusion matrix  Description automatically generated** |
| **Figure 5.1.1 RF w/o Feature Selection** | **Figure 5.1.2 RF w Feature Selection** |

Based on the confusion matrix in Figure 5.1.1 and 5.1.2, the performance of Random Forest without feature selection performed better in predicting the classes of students overall, however the model with feature selection performed better in predicting the ‘High’ class. Based on the result above, we conclude that the model performed better in real-life situations overall is Random Forest without feature selection.

**6.0 Deployment (Contribution by Teh Shu Hui)**

**6.1 Integration into MOOC Platform**

After the model is developed and validated, it can be integrated into the existing MOOC platform. This integration involves embedding the model into the platform's architecture to enable real-time predictions. To enable the integration, we need to create an API to connect the predictive model with the MOOC platform. The API will handle the communication between the platform and the model, passing data back and forth and receiving predictions. The API allows for real-time data preprocessing, where the model can assess student interactions, generating immediate insights. After the API development, we can enhance our user interface with an instructor dashboard, where they can view predictions and insights about student performance. The dashboards could include visualizations, alerts for at-risk students, and recommendations for interventions. Similarly, students can receive personalized feedback based on their predicted performance, including suggestions for additional resources, study strategies, or reminders for upcoming assignments.

**6.2 Monitoring and Maintenance Plan**

Once deployed, the model and its integration with the MOOC platform must be continuously monitored and refined to ensure it remains effective over time. First, we can have performance monitoring by tracking the model’s performance metrics like accuracy, precision, recall to ensure it continues to deliver reliable predictions. An alert system can be set up to notify the development or data science team when performance metrics fall below a certain threshold, indicating potential issues that need immediate attention. Data drift occurs when the statistical properties of the input data change over time, which can degrade the model’s performance. Therefore, continuously monitor the input data distribution and compare it with the training data distribution. Tools like Data Version Control (DVC) can be used to track changes in data distribution. Implementation of drift detection such as Population Stability Index (PSI), to quantify the extent of data drift. If these metrics exceed certain thresholds, it may indicate that the model needs retraining. We can also gather feedback from both instructors and students to identify areas for improvement in the model and the overall user experience. Regularly distribute surveys or feedback forms to instructors and ask for their insights on the model’s predictions, such as the accuracy of at-risk student identification or the usefulness of personalized learning recommendations.

**7.0 Conclusion (Contribution by Chin Wai Teng)**

This research describes a comprehensive method to improve the evaluation and support mechanisms in Massive Open Online Courses (MOOCs) using big data analytics and artificial intelligence (AI). As MOOCs become more popular as a flexible and accessible type of education, substantial issues arise in correctly monitoring student performance and giving timely support to those in need. Traditional assessment methods, which were primarily intended for classroom settings, are frequently insufficient for the diversified and large-scale nature of MOOCs.

To address these issues, this study used a dataset produced from a Learning Management System (LMS) that contains a variety of student features such as demographics, academic activities, and performance measures. To ensure data quality and consistency, the dataset was rigorously prepared using data exploration and preprocessing techniques such as missing value handling, categorical variable encoding, and skewed feature transformation. The report’s main focus is on using multiple machine learning models to predict student performance in MOOCs, including Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM). Following extensive model training and validation, the Random Forest model, with and without feature selection, displayed superior accuracy and consistency. The model’s performance was further optimized via hyperparameter tuning, with the Random Forest model without feature selection emerging as the top overall performer, with an accuracy of 81.25% on the test set. In addition, significance of feature selection in increasing model efficiency and decreasing overfitting. The Variance Threshold approach was used to remove low-variance features, which, while useful for model simplification, had no meaningful impact on the performance of the Logistic Regression and SVM models.

Following the successful construction and validation of the predictive model, the study describes a method for incorporating it into existing MOOC platforms. This integration entails developing an API to enable real-time data processing and prediction, providing immediate insights into student performance. Moreover, is creating an instructor dashboard to track student progress and identify at-risk pupils, as well as offering personalized feedback based on expected performance. The deployed model must be continuously maintained and monitored in order to guarantee its long-term efficacy. In order to improve the model and make it more suitable for changing educational settings, it is necessary for frequent performance monitoring, data drift detection, and user input.

**References**

[1] “What is a MOOC (massive open online course)?,” *WhatIs.com*. https://www.techtarget.com/whatis/definition/massively-open-online-course-MOOC

‌[2]“4 Challenges in Evaluating Massive Open Online Courses (MOOCs),” *Explorance*. https://explorance.com/blog/4-challenges-evaluating-massive-open-online-courses-moocs/

‌[3]R. Baker, P. Mitros, B. Goldberg, and R. Sottilare, “CHAPTER 8 -Assessing Individual Learner Performance in MOOCs.” Available: <https://learninganalytics.upenn.edu/ryanbaker/GIFTCh8.pdf>

[4]“Why and how to use Google Colab | TechTarget,” *Enterprise AI*. <https://www.techtarget.com/searchenterpriseai/tutorial/Why-and-how-to-use-Google-Colab#:~:text=Google%20Colaboratory%2C%20or%20Colab%2C%20is>

[5]teradata, “What Is Python? | Teradata,” *www.teradata.co.uk*. <https://www.teradata.com/Glossary/What-is-Python>

[6]“What Is Matplotlib In Python? How to use it for plotting?,” *ActiveState*. <https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/>

[7]“What is Scikit-Learn In Python?,” *ActiveState*. <https://www.activestate.com/resources/quick-reads/what-is-scikit-learn-in-python/>

[8]NumPy, “What is NumPy? — NumPy v1.19 Manual,” *numpy.org*, 2022. <https://numpy.org/doc/stable/user/whatisnumpy.html>

[9]W3schools, “Pandas Introduction,” *www.w3schools.com*. <https://www.w3schools.com/python/pandas/pandas_intro.asp>

‌

‌

‌

‌