International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 6
ISSN: 2088-5334

Leveraging Data Lake Architecture for Predicting Academic Student Performance

Shameen Aina Abdul Rahim [a], Fatimah Sidi [a,*], Lilly Suriani Affendey [a], Iskandar Ishak [a], Appak Yessirkep Nurlankyzy [b]

[a] *Department of Computer Science, Faculty of Computer Science and Information Technology, University Putra Malaysia, Serdang, Malaysia*
[b] *Department of Computer Science, Faculty of Information Technologies, L. N. Gumilyov Eurasian National University, Kazakhstan*
Corresponding author: [*]fatimah@upm.edu.my

***Abstract*—In today's rapidly evolving landscape of higher education, the effective management and analysis of academic data have become increasingly challenging, particularly in the context of the 3Vs of Big Data: volume, variety, and velocity. The amount of data produced by educational institutions has increased dramatically, including student records. This flood of data originates from various sources and takes several forms, such as learning management systems and student information systems. Hence, in education, data analytics and predictive modeling have become increasingly significant in acquiring insights into student performance, such as identifying at-risk students who are most likely to fail their courses. This study proposes a novel approach for predicting student academic performance, particularly identifying at-risk students, by leveraging a data lake architecture. The proposed methodology comprises the ingestion, transformation, and quality assessment of a combined data source from Universiti Putra Malaysia's Student Information System and learning management system within the data lake environment. With its parallel processing capabilities, this centralized data repository facilitates the training and evaluation of various machine learning models for prediction. In addition to forecasting the student performance, appropriate machine learning algorithms such as Support Vector Classifier, Naive Bayes, and Decision Trees are used to build prediction models by using the data lake's scalability and parallel processing capabilities. This study has laid a solid groundwork for using data architecture to improve students' performance.**

***Keywords*— Data analytics; predictive modeling; student performance; data lake; machine learning algorithms.**

Manuscript received 14 Dec. 2023; revised 19 Mar. 2024; accepted 22 Aug. 2024. Date of publication 31 Dec. 2024.

I. INTRODUCTION

The landscape of higher education is undergoing a transformative shift. Universities are no longer solely focused on imparting knowledge; they increasingly strive to become data-driven institutions that leverage information to optimize student success. This shift is fueled by the Big Data phenomenon: the ever-growing volume, variety, and velocity of data generated by e-learning platforms and other digital touchpoints within the educational ecosystem [1]. These vast datasets hold tremendous potential for understanding student behavior, identifying at-risk students, and improving learning outcomes.

Universiti Putra Malaysia (UPM), like many universities, faces significant challenges in managing and analyzing academic data due to its fragmented nature. Educational data resides in silos across various systems, such as the Student Information System (SIS) and the e-learning platform PutraBlast. This data fragmentation hinders comprehensive analysis and the construction of robust predictive models. SIS typically contained crucial information like grades and course enrollment data, while PutraBlast might hold attendance records and student engagement metrics. This decentralization prevents UPM from gaining a holistic view of student performance. Imagine understanding a student's success based on isolated information scattered across different systems. Identifying trends, patterns, or areas for improvement without a unified view is challenging.

Traditional data management methods create limitations that impede universities' ability to harness modern analytics to predict student outcomes and provide timely interventions [2]. Spreadsheets, for instance, are prone to human error and inconsistencies in data entry and formatting. Additionally, these tools cannot seamlessly integrate data from multiple sources, leading to further fragmentation and hindering the creation of a unified view of student performance. UPM's reliance on traditional data management methods, such as Excel spreadsheets, further increases data fragmentation and

unreliability. This outdated approach impedes the university's ability to harness modern analytics to predict student outcomes and provide timely interventions. As a result, UPM faces difficulties in identifying at-risk students and offering targeted support, ultimately limiting its ability to optimize student outcomes and achieve academic excellence. Implementing a Data Lake architecture could address these issues by creating a centralized, scalable, and flexible data repository, enabling more effective data management and analysis [2], [3].

UPM requires effective methods for identifying at-risk students early to optimize student success. This research proposed a novel approach that leverages the Data Lake architecture to build a comprehensive student performance prediction model. By centralizing academic data from various sources, including the Student Information System (SIS) and the learning management system (LMS), the Data Lake overcomes the limitations of fragmented data [2], [4]. The prediction model utilizes techniques like data preprocessing, feature engineering, and machine learning algorithms that analyze this integrated data to identify students at risk of failing a course at an earlier stage.

An educational data lake management approach was developed to address this issue [5]. Their methodology involves dividing the data lake into three tiers: landing, standing, and consumption, amplified by data models, including Common Data Model (CDM) step Ontology-driven modeling. This approach, which is based on structure, seeks to address the historical challenges of data analysis and strengthen one's capacities for detailed consideration of the entire body of information within an education setting. The application of this approach in the university case study lacks the capacity for providing historical analysis and complementing advanced demonstrates encouraging outcomes, validating properly orchestrated data analytical skills based on rich, actionable information to both educational decision-makers.

Exploring the prediction of academic student performance in e-learning environments, especially those specializing in engagement with the getting-to-know management and student information structures at UPM [6]. The study employs a comprehensive methodology involving data collection, preprocessing, feature engineering, and model training and evaluation. Machine learning algorithms are applied to assess predictive models primarily based on key metrics like accuracy, precision, recall, and the F1-score [7]. Two experiments are performed to investigate the optimal machine learning model and determine the earliest stage for correctly predicting at-risk students. The results highlight the gradient boosting classifier (GB) as the most effective model and become aware of the W1—W12 stage as suitable for early predicting at-risk students. These findings keep implications for timely interventions and support to enhance student achievement in e-learning settings, contributing valuable insights to the evolving environment of virtual education.

Shifting the focus to the healthcare sector [8], present the HEALER architecture, a comprehensive data lake approach designed specifically for healthcare data. Through meticulously evaluating each phase, the researchers provide insightful recommendations for improvement. Noteworthy findings include the efficiency gains associated with larger batch sizes in data ingestion and the recommendation for asynchronous processing of larger datasets to enhance overall system performance. Additionally, the integration of GPU-accelerated libraries, exemplified by KeyBERT, emerges as a potential enhancement for keyword extraction. This study underscores the nuanced considerations required for tailoring data lake architectures to specific domains, showcasing the intricacies of optimizing performance in healthcare data environments.

A comprehensive evaluation of data intake tools for Link Visualizer, a SaaS solution created by Senseworks for the audit sector [9]. This study focused on data pipeline design for audit analytics, which identified possible contenders for the best data extraction tool based on specifications like scalability [10], maintainability, and supportability by combining assessment research with literature investigation. The study technique includes acquiring material from non-traditional sources such as articles, blog posts, and forums since there is a lack of academic research on this rapidly developing subject of data ingestion tools. The research was thought to benefit from experience-based knowledge, and Architecture Decision Records (ADRs) were used to improve the likelihood of choosing the best option. The Proof of Concept (PoC) was constructed using the Airbyte Open-Source framework, and part of the evaluation process involved building an ADR to select data intake methods. While it did not promise a flawless answer, the Proof of Concept showed where there was room for growth in terms of fulfilling the requirements. In addition to actual community comments from sites like stackoverflow.com, reddit.com, and medium.com, Senseworks exploited its in-use expertise with the current data integration solution to gain significant insights into the requirements and obstacles for a new tool. With further work involving the creation of other PoCs for useful comparisons and assuring solution quality through rigorous testing, the successful integration of Airbyte in the PoC demonstrated its potential to improve the data collection procedure for Link Visualizer.

E-learning and predictive analytics were conducted, and the methodology involved a comprehensive process, starting from data collection from the Open University Learning Analytics Dataset (OULAD), preprocessing, feature selection, model development, and evaluation [11]. The outcome is developing a predictive model that effectively forecasts students' performance in e-learning environments based on their learning process and behavior data. This model holds significant promise, potentially assisting educators in identifying students at risk of poor performance and facilitating targeted interventions to improve learning outcomes. The study highlights the transformative power of predictive analytics in shaping personalized and effective educational interventions in the context of e-learning [10], [12].

This paper delves into the existing literature on student performance prediction in Section 1, outlines the implementation of the proposed model within UPM's Data Lake architecture in Section 2, analyzes the results and their implications for the university in Section 3, and concludes with key findings and future directions in Section 4.

II. Materials and Method

This section outlines the detailed material and method for forecasting student success at UPM. The method was carried out to effectively identify and support students who might be struggling early in their studies, ensuring they receive help on time. By using a varied set of data gained from the SIS and PutraBlast, the main goal of this study is to check the possibilities of students who 'pass' and might be 'at-risk'. This includes the preparation of data and detailed analysis of data from the SIS and PutraBlast, integrated within a data lake architecture proposed in Fig. 1, which adopts a well-organized Data Lake framework, as described in [10] and incorporates the Zone Architecture from [13].

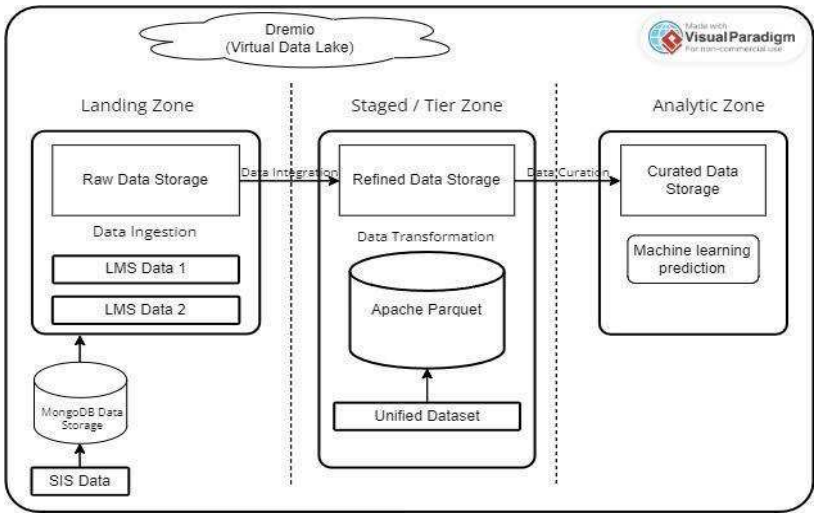

Fig. 1 Design of Centralized Data Lake Architecture

This framework emphasizes the importance of data organization, ensuring the risk of the database becoming a mess where data becomes difficult to use is reduced:

1) Landing Tier: This is the initial phase of our Data Lake, similar to the one described in [10]'s model. It is the gateway for all incoming raw data, no matter its initial structure. Information from various internal sources at UPM, such as the SIS and PutraBlast platform, will be stored in this tier without any modifications. This method guarantees that all accessible data is captured and available for subsequent analysis.

2) Staging Tier: The second tier is dedicated to transforming and cleaning the raw data ingested from the landing tier. Here, the data will undergo critical processes to ensure consistency and quality. This may involve addressing missing values and correcting inconsistencies in data formats.

3) Consumption Tier: This tier is the repository for all the prepared and organized data. Here, the data will be readily accessible for predictive analytics tasks.

The creation of features and the use of modern machine learning techniques such as Support Vector Classifier (SVC), Naive Bayes, and Decision Trees [6], [14] are crucial in predicting student performance that is not only accurate and efficient but also plays a key role in supporting early interventions designed to improve student retention and promote academic achievement.

A. Dataset

This study derived a dataset from previous research by [6] encompassing student performance data for undergraduate students in the UPM Faculty of Computer Science and Information Technology (FSKTM) during the 2020/2021 academic session (two semesters). The dataset includes information for 705 students across 32 courses and comprises 2416 data points. Data relevant to the student performance prediction model was extracted from the two university systems: The student Information System (SIS) and the learning management system (PutraBlast).

The SIS data provided student demographics included student ID (unique identifier, String), gender (String), age (years, Integer), marital status (String), country of origin (String), and sponsorship type (String) with no missing values. However, to optimize the prediction model attributes like faculty, semester, student ID, course name, and grades were excluded during preprocessing, reducing the initial set of 21 features to 16. PutraBlast data consisted of raw event logs capturing student interactions such as accessing course materials and submitting assignments. These interactions, categorized as "course viewed," were tracked weekly throughout the semester (Week 1 to Week 19). Four cumulative course view variables were constructed based on the weekly data for prediction stages at Weeks 7, 12, 14, and 19 to reflect varying course lengths and facilitate predictions at different stages.

TABLE I
Grading Scheme Used in the Dataset by [6]

Marks	Grades	Value Point
80-100	A	4.000
75-79	A-	3.750
70-74	B+	3.500
65-69	B	3.000
60-64	B-	2.750
55-59	C+	2.500
50-54	C	2.000
47-49	C-	1.750
44-46	D+	1.500
40-43	D	1.000
39 or less	F	0.000

Finally, student performance prediction is framed as a binary classification problem with two classes: "pass" and "at-risk." The target class designation ("pass" or "at-risk") was derived from the grade data based on UPM's grading scheme (details provided in Table I of the original paper by [6]). Grades C and below were classified as "at-risk" to identify students in danger of failing and enable early intervention, aiming to prevent potential academic probation due to low GPA and CGPA.

B. Data Storage

This section presents MongoDB, a Non-Relational (NoSQL) database that excels in managing and storing information [15] typically found within a SIS. Normally, data from SIS is exported in a recognizable format – Excel files (.xlsx). However, these spreadsheets must be securely stored and managed before they can be utilized. This is where MongoDB comes into play. Transferring Excel file data into MongoDB involves several critical steps to ensure accurate data transformation and secure storage, which are then accessible for further analysis with Dremio.

This phase may require writing scripts or using SIS tools to export the data in a format MongoDB can accept. It often changes it to a more flexible format like CSV or JSON, which is easier to manipulate programmatically. Excel data may not easily fit into MongoDB's document-oriented structure. At this point, the data needs to be prepared for storage within MongoDB documents. This preparation includes several tasks:

Each column from the Excel file is linked to a matching field in MongoDB's document schema. For example, fields like "Student ID," "Name," "Age," "Gender," "Major," and "Enrollment Status" from the Excel file are converted into fields in a MongoDB document. This conversion step guarantees that MongoDB's hierarchical and flexible document structure is effectively utilized. Any gaps in data, inconsistencies in the Excel files, or issues in data type must be addressed before the data is uploaded to MongoDB. This may involve filling in missing data, correcting errors in data entry, or changing data types to ensure consistency throughout the dataset.

Since MongoDB stores data in a document-based format, the extracted and pre-processed Excel data must be transformed into JSON or BSON format. Each row in the Excel file is turned into a MongoDB document, with each cell serving as a field-value pair. This transformation process ensures that the data maintains its structure and is readily available in MongoDB.

The subsequent phase involves inserting it into MongoDB storage. This is achieved through MongoDB's import functionality or its Application Programming Interface (API). The converted documents are then placed in a MongoDB database, serving as a student demographic information repository. This database offers a flexible and efficient way to store vast amounts of data and perform intricate searches. After the data is uploaded into MongoDB, it is crucial to carry out a verification process to ensure its accuracy and thoroughness. This step requires a query of the MongoDB database to compare the output with the initial Excel data. Any inconsistencies are resolved to verify that the data has been successfully imported.

C. Data Ingestion

This study uses Dremio, a data lake platform, to simplify the process of integrating student information from PutraBlast and SIS systems. Dremio acts as a bridge between these diverse data sources [16], which in this case are Excel files (.xlsx). This platform is cleverly placed inside Docker containers to make getting student data from PutraBlast and SIS systems more streamlined. Docker serves as a platform for software containerization, providing an attractive method for deploying and managing Dremio.

Upon identifying the files, Dremio automatically transforms them into a columnar format optimized for efficient data storage and retrieval within its data lake architecture. This columnar format offers significant performance advantages over traditional row-based storage, especially for large student datasets and complex queries.

Next, the data will be parsed within each Excel file, identifying the headers that define each data point's column name. This step ensures accurate data interpretation for future analysis. Dremio will detect the data type using its detection capabilities, which assign appropriate data types (integers, floating-point values, date/time, or strings) to each column. This ensures data consistency and facilitates efficient data manipulation during analysis. Assigning accurate data types is crucial for valid statistical operations on the student information.

In the final stage, virtual datasets will be created on top of the physical Excel files. These virtual datasets act as an abstraction layer that enables users to query the student information using standard SQL syntax. This eliminates the need to know the specific location or format of the original files, simplifying data access and manipulation. Virtual datasets also offer benefits like data security and version control, ensuring data integrity and facilitating collaborative analysis.

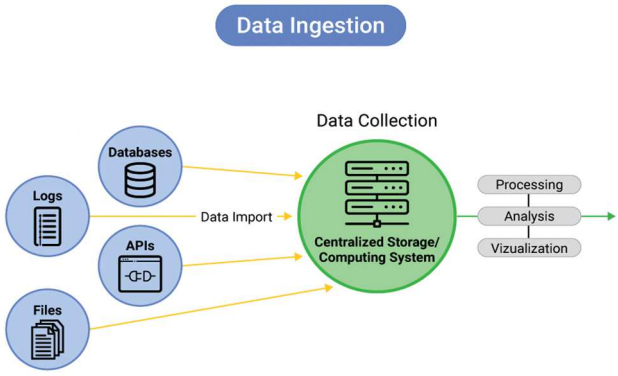

Fig. 2 Flow of Data Ingestion process using Dremio

D. Data Integration

Data ingestion focuses on bringing separate data sources [15] into Dremio, while data integration takes that ingested data and combines it [17], [18]. Data integration aimed to create a unified view [17] of student data by combining relevant information from both systems. Dremio acts as a central hub [16], seamlessly ingesting data from separate sources like PutraBlast and SIS. Data ingestion involves the initial extraction and loading of raw data from these systems into Dremio. This study adopts an ELT (Extract, Load, Transform) approach [19], where the raw data is first loaded

into Dremio without immediate transformation. This approach prioritizes initial data availability for exploration and facilitates efficient use of Dremio's processing capabilities for later data refinement.

Following the initial data load, Dremio's user interface becomes the platform for thorough data integration. Here, a comprehensive examination of the schemas (structures) for both PutraBlast and SIS datasets will be delved into. This examination aims to deeply understand the available data points (columns) and their corresponding data types (categorical or numerical). Additionally, descriptions provided for each variable are meticulously reviewed to grasp the intended meaning and identify any potential inconsistencies within the data. This in-depth exploration ensures a clear understanding of the data landscape and allows seamless data merging.

A critical step in data integration involves establishing a common ground for merging the datasets from PutraBlast and SIS. This research identifies the student identifier as the ideal candidate for the primary join key. Both Putrablast ("STUD_MATRIC_NO") and SIS ("Student ID") likely contain unique identifiers for each student. Since these columns share the same meaning and format (likely strings representing student ID numbers), they were chosen as the primary join key. This selection ensures accurate and efficient data merging. Dremio can merge corresponding student data points from both systems by utilizing a common and unique identifier, creating a unified and comprehensive view of student information within the data lake. This unified view will become the foundation for further analysis and model development to predict student performance and identify at-risk students.

E. Data Transformation

While Dremio might handle basic data cleaning during the ingestion process, further refinement might be necessary. The quality of data is paramount for optimal performance in predictive models. Consequently, data preprocessing is a crucial step in preparing the data for model training and evaluation. This stage transforms the data into a format suitable for the chosen machine learning algorithms [20]. Several techniques are employed here to ensure the data is well-prepared for model development:

1) Normalization: The dataset includes numerical features with different scales, which could lead to a bias towards features with larger values during the model's training. To mitigate this, normalization is applied, a method that adjusts numerical features to have a mean of 0 and a standard deviation of 1 [6], [21]. This study uses the StandardScaler() function from the scikit-learn library to perform normalization. StandardScaler adjusts the data to have a mean of 0 and a standard deviation of 1, ensuring that all features contribute equally to the learning process of the model**.**

*2) Encoding categorical variables***:** Numerous machine learning algorithms have difficulty processing categorical data presented as strings or objects. To circumvent this issue, these features are converted into numerical representations for better understanding by the algorithms. This study employs label encoding, a widely used method that converts each unique category into a corresponding integer label. The LabelEncoder() function from the scikit-learn library is utilized for this conversion. Label encoding assigns a unique integer to each category in a column, effectively transforming categorical data into numerical form. Replacing each categorical value with its assigned integer label preserves the order of the data if it exists, which is especially beneficial when dealing with categorical features with a meaningful order.

*3) Overcoming class imbalance***:** Upon examination of the dataset, it's clear that there is a significant imbalance in the classes, with the "at-risk" student category only making up 7.16% of the dataset, or 2416 rows. This imbalance is partly due to duplicate student IDs, as students might enroll in multiple courses during a semester. Such imbalance can skew the model towards predicting the majority class (non-at-risk students), potentially leading to errors in identifying and supporting at-risk students. To effectively tackle this problem, the dataset will be divided into three subsets through a train-test-validation split. This well-established method allocates the dataset into three distinct subsets: a training set, a validation set, and a test set [22]. This division aims to ensure the model is trained on a balanced representation of both classes (pass and at-risk students) across all three subsets – training, validation, and test data.

F. Feature Extraction

This study adopts the feature extraction procedure recommended by [6], which also referred to other studies' suggested procedures. The method approach divides the course into distinct stages for targeted feature extraction. This multi-granular approach aims to capture student performance and identify at-risk students at various points throughout the semester, enabling earlier intervention strategies. Four distinct stages were defined based on key percentage milestones of course completion. These milestones (e.g., 37%, 63%) were strategically chosen to represent critical junctures in the learning process. The rationale behind these stages is that students' early interactions with course materials and initial performance indicators can provide valuable insights into their potential struggles. Additionally, capturing performance data at later stages allows for incorporating cumulative learning progress.

At each stage, relevant data points are extracted from the SIS dataset. This includes:

1) Demographic Information: Features such as student ID, age, gender, and potentially other relevant demographic data points can offer insights into student characteristics that may influence academic performance [23].

2) Course Information: Course code, name, and relevant course-specific details can be included to capture potential course difficulty or workload variations.

3) Student Percentage Marks (*Stage-Specific*): Extracting student percentage marks achieved up to the designated milestone (e.g., percentage marks for Weeks 1-7 at Stage 1) provides a direct measure of early academic performance.

The feature set is further enriched by integrating course view data from the LMS. This data captures the total number

of times students have accessed course materials within the designated time frame for each stage (e.g., Week 1 to Week 7 for Stage 1, Week 1 to Week 12 for Stage 2, and so on). This information is a proxy for student engagement and interaction with the course content, potentially offering early signs of potential challenges.

G. Data Modeling and Evaluation

During this stage, predictive models will be built using data engineered from earlier steps in data preprocessing and analysis. The models chosen for this process include a mix of machine learning methods, such as Decision Trees (DT), Support Vector Classifiers (SVC), and possibly more, depending on how they are applied. These models are then trained on a segmented dataset by splitting it into three parts: a training set for model training, a validation set for adjusting model settings and choosing the best model, and a test set for evaluating how well the models perform on data.

Every model is trained on a training set with specific settings and parameters tailored to its approach. After the training phase, the models' abilities are thoroughly tested using the validation set [24]. Important measures like accuracy, precision, recall, and F1-score are calculated to determine how well each model can predict student outcomes and identify students who might be at risk based on the features engineering. The selection of appropriate metrics is crucial for effectively evaluating the performance of a predictive model [24], [25]. This study utilizes a combination of commonly employed metrics to comprehensively assess the effectiveness of the developed models in predicting at-risk students.

1) Accuracy: Accuracy, a widely used metric, is calculated by dividing the number of correctly classified instances by the total number of samples in the dataset [6]. While intuitive, accuracy can be misleading in scenarios with imbalanced datasets, where the model might be biased towards the majority class [6]. The formula for accuracy is presented below (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

where:
TP (True Positive) = Correctly predicted positive cases
TN (True Negative) = Correctly predicted negative cases
FP (False Positive) = Incorrectly predicted positive cases (Type I error)
FN (False Negative) = Incorrectly predicted negative cases (Type II error)

2) Precision: Precision focuses on the proportion of truly positive predictions [6]. It is calculated by dividing the number of correctly identified positive cases (TP) by the total number of positive predictions the model makes (TP + FP). The formula for precision is as follows (2).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

3) Recall: Recall, also known as sensitivity, measures the model's ability to identify all actual positive cases [6]. It is calculated by dividing the number of correctly identified positive cases (TP) by the total number of actual positive cases in the dataset (TP + FN). The formula for recall is presented in (3).

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

4) F1-score: The F1-score provides a harmonic mean of precision and recall, offering a more balanced view of the model's performance [6]. It is particularly valuable in imbalanced datasets, where solely relying on accuracy can be misleading. The F1-score is calculated using the following formula (4).

$$F1-score = 2x \frac{Precision\ x\ Recall}{Precision + Recall} \quad (4)$$

III. Results and Discussion

The strategy of train-test-validation split was employed to ensure a balanced and unbiased representation of both student categories ("at-risk" and "pass") across the training, validation, and test datasets. The training set is 70% and the test set is 30%. The validation is using the same size of data. This approach is crucial for preventing bias that could occur if certain student categories were disproportionately represented in specific datasets. By maintaining a balanced distribution, the evaluation of the model's performance on unseen data becomes more reliable and applicable.

The Decision Tree Classifier, set at a maximum depth of 3, achieved high accuracy as in Table II across all datasets: 0.9911 on the training set, 0.9770 on the validation set, and 0.9744 on the test set. This model achieved a precision of 1.00 and a recall of 0.58, resulting in an F1-score of 0.74. The validation set showed improved precision at 0.91 and recall at 0.83, with an F1-score of 0.87. The test set exhibited a balanced performance with a precision of 0.92 and a recall of 0.92, leading to an F1-score of 0.92. This result indicates outstanding performance but also raises concerns about the possibility of overfitting. In this case, the model might have become too proficient in memorizing the training data, possibly including irrelevant or noisy patterns. This could lead to overly positive performance metrics that might not hold up when applied to new, unseen data.

The Support Vector Classifier (SVC) demonstrated a good balance between performance on the training and validation sets, with accuracy scores of 0.9947 and 0.9862, respectively. The test accuracy of 0.9764 further supports this, suggesting that SVC can learn from the training data while still being able to generalize well to new, unseen data in the validation set. The test accuracy of 0.9655 further supports this idea. The performance indicators of the SVC were notably impressive, boasting a precision of 1.00 and a recall rate of 0.94 on the training data, leading to an F1-score of 0.97. When applied to the validation data, it managed a precision of 0.98 and a recall rate 1.00, achieving an F1-score of 0.99. The performance on the test data mirrored this consistency, with precision and recall reaching 1.00 and an F1-score of 1.00. This uniformity underscores this model's ability to accurately pinpoint the best hyperplane that maximizes the separation between the two student categories ("at-risk" and "pass"), rendering it a strong predictor of student outcomes.

Gaussian Naive Bayes, a classifier based on the assumption that features are independent within each class, performed exceptionally well across all datasets. The model achieved accuracy scores of 0.9994 on the training set, 0.9908 on the validation set, and 0.9902 on the test set. The evaluation metrics demonstrated exceptional accuracy and the ability to

correctly identify instances, with the training data achieving a precision of 1.00 and a recall of 0.94, leading to an F1 score of 0.97. The evaluation data also displayed an accuracy of 0.99 and recall of 1.00, with an F1-score of 0.99. The testing data continued to perform at this level, with precision and recall at 1.00 and an F1-score of 1.00. These outcomes indicate the model's success in categorizing student information and its robustness in applying it to new data. However, it's essential to consider the assumption that features are independent, as this may not apply universally to all datasets, which could impact the model's performance across various situations.

Logistic Regression, designed for binary classification tasks, was used to predict the probability of a student being at risk based on their features [26], [27], [28]. The model provided probability scores for each class, and its accuracy was compared with the other models. Logistic Regression achieved training, validation, and test accuracies of 0.9397, 0.9401, and 0.9350, respectively. In the training data, the model achieved a precision of 0.48 and a recall of 1.00, resulting in an F1 score of 0.65. The data used for validation demonstrated an accuracy of 1.00 and recall of 0.94, with an F1-score of 0.97. The data from the test set also showed consistent performance, with both precision and recall values at 0.92, leading to an F1-score of 0.92. These findings indicate that although Logistic Regression was effective, it fell short of fully understanding the intricate connections between student characteristics and their risk levels compared to models like SVC or Gaussian Naive Bayes.

TABLE II
COMPARISON OF ACCURACY BETWEEN EACH MODEL

Model	Training Accuracy	Validation Accuracy	Test Accuracy
Decision Tree	0.9911	0.9770	0.9744
Support Vector Classifier (SVC)	0.9947	0.9862	0.9764
Gaussian Naive Bayes	0.9994	0.9908	0.9902
Logistic Regression	0.9397	0.9401	0.9350

After conducting a detailed comparison in Table II and choosing the best model for predicting at-risk students, valuable insights were gained by looking at how well each model did on training, validation, and test data. The Decision Tree showed impressive accuracy during these evaluations, which made fitting the data too closely and looked further into its ability to work well with new data. On the other hand, the Support Vector Classifier (SVC) came out as a strong contender, performing consistently well across all datasets and proving itself good at making predictions.

The Gaussian Naive Bayes model did remarkably well, especially because it matched the idea that each class has independent features. However, it's important to remember how true this is and how well it works in situations where this independence might not be accurate. Similarly, Logistic Regression did a good job. Still, its lower accuracy levels suggest it might struggle to understand the full depth of the relationships more than some other models in our study.

Considering these results, the SVC model is the best option for predicting at-risk students in this situation because it performs well overall and finds the best line for separating classes. Still, there is a need to explore the other models more, especially on preventing Decision Tree overfitting and looking more closely at how the Gaussian Naive Bayes model treats features independently in each class to improve how well it works in different situations. Thus, future research could greatly benefit from a closer look at these aspects to make better choices for predicting outcomes in educational settings.

The study by [8] used a traditional approach, relying on Excel documents to analyze data instead of a data lake framework. In [8], it was found that early prediction models were viable when certain performance indicators like accuracy, precision, recall, and F1 scores were between 75% and 85%. Conversely, this study achieved impressive outcomes with the Gradient Boosting model, showing varying performance levels throughout the course. When predicted using complete course data, the GB model reached its peak performance in precision, recall, and F1 scores (91.79%, 98.48%, and 94.84%, respectively). However, the model's performance dropped when predicted from earlier course stages, such as W1—W14, experiencing a precision score decrease of 33.1%. The SVC model also performed well, with precision scores of 90.35%, recall scores of 96.93%, and F1-scores of 93.55% for the full course length, though it saw a decrease in performance in earlier stages. Despite these drops, the W1—W14 stage was the closest to meeting the performance benchmarks set by [6], achieving a recall score of 83.5%. This underscores the crucial role of final exam grades in predicting outcomes and the benefits of employing a data lake framework for more comprehensive data analysis and early intervention strategies.

IV. CONCLUSION

This study highlights the significant obstacles Universiti Putra Malaysia (UPM) encounters in effectively managing and analyzing educational data. The disjointed and inconsistent spread of data across various university systems [29], [30], such as the Student Information System (SIS) and the Learning Management System (PutraBlast), severely limits UPM's capacity to obtain a full understanding of student achievement. This absence of a unified data storage system adds complexity, making it difficult for the university to extract valuable insights and provide timely support to students who might be struggling. Furthermore, the scattered nature of educational data complicates efforts to ensure data security and privacy, as confidential student information is spread across different platforms, heightening the risk of unauthorized access and data leaks. This situation threatens the accuracy of student records and raises questions about adherence to data protection laws.

A solution has been proposed to tackle these pressing issues. This solution consolidates various educational data sources into a unified data framework using advanced data management technologies, especially by utilizing a data lake environment supported by Dremio. By adopting this strategy, the data ingestion and transformation processes will become more efficient, significantly improving data management within the university. This, in turn, will facilitate a more precise predictive analysis of student success, ultimately benefiting the entire UPM community.

Adopting this suggested solution will not only help UPM address its current challenges in managing educational data but will also set the university up for future success in using data-driven insights to improve student outcomes and overall academic quality. By adopting modern data management technologies, UPM can ensure its leadership in innovation and educational excellence in Malaysia and beyond. Recognizing the impact of data on student performance and the overall quality of education is crucial for creating an environment that supports ongoing improvement and personalized learning experiences. By understanding the obstacles faced by universities like UPM and the proposed solutions, students can better appreciate the efforts to enhance their educational journey and contribute to a more data-aware educational environment.

This study has laid a solid groundwork for using data architecture to improve students' performance. However, there are many exciting areas for further study and innovation. A major focus is making the most of the data lake by using advanced analytics methods. Adding complex machine learning algorithms and AI models could significantly enhance the system's ability to predict student success [7]. By exploring detailed patterns in the data, researchers can create models that accurately identify students who might struggle early on, allowing for prompt support and customized help.

Developing effective data visualizations and user-friendly dashboards is essential for sharing insights from the data with various stakeholders, including educators, administrators, and students. Creating interactive visual tools helps in understanding trends in student performance and identifying strengths and areas needing improvement, which supports making decisions based on data. Ensuring its privacy, accuracy, and safety is a top priority when managing and securing student data. Implementing strict data management policies and security measures is key to protecting sensitive information. Regular data security checks and reviews are necessary to meet relevant standards and build trust within the university community.

Growing UPM's data collection by incorporating various sources can provide a more comprehensive and detailed view of student achievement. By combining information from systems like student feedback, alumni surveys, or external evaluations, the analysis can be more thorough, leading to more accurate predictions and useful insights. By focusing on these important areas, UPM is well-positioned to fully leverage data in making decisions, ultimately creating a more personalized and effective educational experience for its students.

ACKNOWLEDGMENT

The Faculty of Computer Science and Information Technology, University Putra Malaysia, funded this research.

REFERENCES

[1] H. Jahankhani, A. Jamal, G. Brown, E. Sainidis, R. Fong, and U. J. Butt, Eds., AI, Blockchain and Self-Sovereign Identity in Higher Education. Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-33627-0.

[2] R. Raju, R. Mital, and D. Finkelsztein, "Data Lake Architecture for Air Traffic Management," *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pp. 1–6, Sep. 2018, doi:10.1109/dasc.2018.8569361.

[3] P. Wieder and H. Nolte, "Toward data lakes as central building blocks for data management and analysis," *Frontiers in Big Data*, vol. 5, Aug. 2022, doi: 10.3389/fdata.2022.945720.

[4] A. Cuzzocrea, "Big Data Lakes: Models, Frameworks, and Techniques," *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 1–4, Jan. 2021, doi:10.1109/bigcomp51126.2021.00010.

[5] D. Martinez-Mosquera, V. Beltran, D. Riofrio-Luzcando, and J. Carrion-Jumbo, "Data Lake Management for Educational Analysis," *2022 IEEE Sixth Ecuador Technical Chapters Meeting (ETCM)*, pp. 1–5, Oct. 2022, doi: 10.1109/etcm56276.2022.9935751.

[6] S. M. M. Muin et al., "Predicting academic student performance based on e-learning platform engagement using learning management system data," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 9, pp. 1859–1866, 2023. doi: 10.17762/ijritcc.v11i9.9178.

[7] H. S. Brdesee, W. Alsaggaf, N. Aljohani, and S.-U. Hassan, "Predictive Model Using a Machine Learning Approach for Enhancing the Retention Rate of Students At-Risk," *International Journal on Semantic Web and Information Systems*, vol. 18, no. 1, pp. 1–21, Mar. 2022, doi: 10.4018/ijswis.299859.

[8] C. Manco, T. Dolci, F. Azzalini, E. Barbierato, M. Gribaudo and L. Tanca, "HEALER: A data lake architecture for healthcare", *Proc. Workshops EDBT/ICDT Joint Conf.*, pp. 1-10, 2023, Mar. 19, 2024, [online] Available: https://ceur-ws.org/Vol-3379/DataPlat_2023_602.pdf.

[9] G. Weintraub, E. Gudes, S. Dolev, and J. D. Ullman, "Optimizing Cloud Data Lake Queries With a Balanced Coverage Plan," *IEEE Transactions on Cloud Computing*, vol. 12, no. 1, pp. 84–99, Jan. 2024, doi: 10.1109/tcc.2023.3339208.

[10] D. Mazumdar, J. Hughes, and J. B. Onofre, "The Data Lakehouse: Data Warehousing and More," 2023. [Online]. Available: arXiv:2310.08697. [Accessed: Mar. 19, 2024].

[11] F. Qiu et al., "Predicting students' performance in e-learning using learning process and behaviour data," *Scientific Reports*, vol. 12, no. 1, Jan. 2022, doi: 10.1038/s41598-021-03867-8.

[12] J. Fan, "A big data and neural networks driven approach to design students management system," *Soft Computing*, vol. 28, no. 2, pp. 1255–1276, Dec. 2023, doi: 10.1007/s00500-023-09524-8.

[13] R. P. d. C. C. Macedo, "Implementation of a Data Lake in a Microservices Architecture," Master dissertation, Department Information, University of Lisbon, Portugal, 2024, [Online] Available: http://hdl.handle.net/10451/63925.

[14] R. Asokan, D. P. Ruiz, and S. Piramuthu, Eds., Smart Data Intelligence. Springer Nature Singapore, 2024. doi: 10.1007/978-981-97-3191-6.

[15] "What is a non-relational database?" (n.d.). [Online]. Available: https://www.mongodb.com/resources/basics/databases/non-relational. Accessed: Mar. 19, 2024.

[16] "What is a cloud data lake?" (n.d.). [Online]. Available: https://www.dremio.com/resources/guides/cloud-data-lakes/. Accessed: Sept. 24, 2024.

[17] C. Cuello, "Data Ingestion vs. Data Integration: Know the differences for efficient data management," Dec. 5, 2023. [Online]. Available: https://rivery.io/data-learning-center/data-ingestion-vs-data-integration/. [Accessed: Mar. 19, 2024].

[18] "Data Ingestion vs. Data Integration: What Sets Them Apart?" Feb. 27, 2024. [Online]. Available: https://airbyte.com/data-engineering-resources/data-ingestion-vs-data-integration. Accessed: Mar. 19, 2024.

[19] M. Garcia, "The Evolution of Data Pipelines: ETL, ELT, and the Rise of Reverse ETL," CORE, Oct. 2, 2023. [Online]. Available: https://dzone.com/articles/the-evolution-of-data-pipelines.

[20] "What is Data Transformation?" (n.d.). [Online]. Available: https://www.tibco.com/glossary/what-is-data-transformation#, Accessed: Mar. 19, 2024.

[21] D. Ushasree, A. V. Praveen Krishna, and Ch. Mallikarjuna Rao, "Enhanced stroke prediction using stacking methodology (ESPESM) in intelligent sensors for aiding preemptive clinical diagnosis of brain stroke," *Measurement: Sensors*, vol. 33, p. 101108, Jun. 2024, doi:10.1016/j.measen.2024.101108.

[22] J. Xiong et al., "Deep Learning-Based Open Source Toolkit for Eosinophil Detection in Pediatric Eosinophilic Esophagitis," Aug. 2023. [Online]. Available: https://arxiv.org/abs/2308.06333.

[23] G. Siemens, "Learning Analytics," *American Behavioral Scientist*, vol. 57, no. 10, pp. 1380–1400, Aug. 2013, doi:10.1177/0002764213498851.

[24] "Streamlining Predictive Analytics with Scikit-Learn," (n.d.). [Online]. Available: https://www.osedea.com/insight/streamlining-predictive-analytics-with-scikit-learn. Accessed: Mar. 19, 2024.
[25] N. Sghir, A. Adadi, and M. Lahmer, "Recent advances in Predictive Learning Analytics: A decade systematic review (2012–2022)," *Education and Information Technologies*, vol. 28, no. 7, pp. 8299–8333, Dec. 2022, doi: 10.1007/s10639-022-11536-0.
[26] "Model Performance," (n.d.). [Online]. Available: https://fastercapital.com/keyword/model-performance.html. Accessed: Sept. 24, 2024.
[27] Z. Liu, W. Chen, C. Liu, R. Yan, and M. Zhang, "A data mining-then-predict method for proactive maritime traffic management by machine learning," *Engineering Applications of Artificial Intelligence*, vol. 135, p. 108696, Sep. 2024, doi: 10.1016/j.engappai.2024.108696.
[28] D. T. Larose, "Data Mining Methods and Models," John Wiley & Sons, Inc, Nov. 2005, doi: 10.1002/0471756482.
[29] P. Pooja and R. Bhalla, "A Review Paper on the Role of Sentiment Analysis in Quality Education," *SN Computer Science*, vol. 3, no. 6, Sep. 2022, doi: 10.1007/s42979-022-01366-9.
[30] P. Rangnekar, "What is Mobility in App Development? Key Insights!," Jan. 2024. [Online]. Available: https://www.biz4solutions.com/blog/category/uncategorized/page/13/. Accessed: Mar. 19, 2024.