Predicting Students Performance In Exams Using Machine Learning Techniques

Khanyisile Sixhaxa
School of Computer Science and Applied Mathematics
The University of the Witwatersrand
Johannesburg, South Africa
1590202@students.wits.ac.za

Ashwini Jadhav
Science Teaching and Learning Unit
Faculty of Science
The University of the Witwatersrand
Johannesburg, South Africa
ashwini.jadhav@wits.ac.za

Ritesh Ajoodha
School of Computer Science and Applied Mathematics
The University of the Witwatersrand
Johannesburg, South Africa
ritesh.ajoodha@wits.ac.za

***Abstract*—Predicting students success has been a very popular study across different fields and with this study we will be focusing on how Machine Learning can aid us in giving us insight in how students will perform in their exams. This paper will present a study on which Demographic, Academic and Behavioural features play a significant role in how a student performs in exams with an added feature which looks into the student's family background also. For the analysis of the features we will use the Mutual Information algorithm, alongside five machine learning models that which are a mixture of classification and regression classifiers. We will also use these models to predict our students performance. The five classifiers used in this study are as follows: Gaussian Naïve Bayes, Support Vector Machine, Random Forest, K-Nearest Neighbour and Logistic Regression and we achieved prediction accuracy of 50.83%, 81.67%, 78.33%, 75.00% and 74.17% respectively.**

I. Introduction

Measuring of a students academic performance has been a topic of interest across various fields. This study proves itself to be a challenge due to the fact that students academic performance is reliant on various factors like behavioural, academic, demographics and other environmental factors. The purpose of this paper is to predict the students performance in exams and the factors that influence their performance.

Machine learning is a field in which there has been a success in being able to predict students performance, this is due to the fact that there are algorithms that have been developed to help achieve better accuracy on predictions and also being able to realize important attributes that play a critical role.

When trying to predict the likelihood of certain events, we look into using supervised classification machine learning algorithms. In this study we will employ five classification algorithms: Gaussian Naïve Bayes, K-Nearest Neighbour, Support Vector Machines, Logistic Regression and Random Forest.

For this study we will use the x-API Edu data set which was obtained from the Kalboard 360 Learning Management System (LMS). Due the high dimensionality nature of our data set, it is inevitable that we will get low accuracy from our prediction models. To better our prediction accuracy, we will have to pre-process our data to reduce the bias and variance we get from our models.

We can then perform feature selection on our data set, this will help us see the features with the most variance. This will enable us to correlate which attributes are detrimental to a students performance. We can then explore how we can help students improve their academic performance accordingly

II. Related Work

Predicting student performance using data mined from a Learning Management System can help us identify and help at risk students [1]–[3]. In [2], the authors used data mining techniques to get get attributes about students to put together the x-API Edu Data set. We see the use of the data mining in [4]–[6] which were mostly mined from data mining learning management system.

The correlation between students performance against their demographic, behavioural and academic attributes has been explored extensively in [3], [4], [7] and we see that behavioural attributes play a more critical role in a students performance. We see the use mutual information/ information gain for feature selection in [1]–[3] as it calculates the importance of of the independent feature against dependent feature and allocates an entropy score between 0 and 1. This allows us to rank the features according to their importance.

For our prediction models, we see that the models used in this paper are great when dealing with complex data sets with high dimensionality [4], [6], [8]. These can help us predict at risk students with high accuracy and low computational costs. With Random Forest, Naïve Bayes, Logistic Regression and Support Vector Machines achieving 73.19%, 72.44%, 72.39% respectively in [1].

III. Methodology

A. Data Source

In this study we used the Students' Academic Performance data set (xAPI-Edu-Data) obtained from Kaggle. This is an educational data collected from the Kalboard 360 which is a online Learning Management System (LMS) used across the world. This system serves as an online educational platform that enables educators to share resources, make announcements, hand out assignments, etc. and for students to access the given resources, view the announcements and assignments,

etc. We are able to obtain information about students using a monitoring component of the LMS called the experience API (xAPI). This API tracks a students behaviour and actions, from viewing announcements to reading of available resources.
In this xAPI-Edu-Data, we have 480 data instances with 16 features. The features are partitioned into three categories: Demographic, Academic and Behavioural feature sets. Our target variable is the Class feature, where the students marks can be classified into three categories:

- **Low:** students who achieve 0-69
- **Medium:** students who achieve 70-89
- **High:** students who achieve 90-100

It is worthy to note that our data only has 4 features are numerical and the rest are categorical. Let us have a look into the given features according to their categories in Table I.

TABLE I
LIST OF FEATURES

No.	Demographic Features
1	Gender
2	Nationality
3	Place of Birth
	Academic Features
4	Educational Stages
5	Grade Levels
6	Section ID
7	Topic
8	Semester
9	Parent responsible for student
	Behaivoural Features
10	Raised Hands
11	Visited Resources
12	Viewing Announcements
13	Discussion Groups
14	Parent Answering Survey
15	Parent School Satisfaction
16	Student Absence Days

B. Data Pre-processing

In order for us to use the data, we need the remove the noise in the data so that our models can work efficiently. The term "noise" implies the instances in our data that could lead to distorted results and leaves room for increased error margins. To stabilize our data, we will pre-process our data, this is an extremely crucial step for Machine Learning due to the fact that real-world data consists of incomplete, inaccurate and inconsistent data instances. In this study, to process our data we perform the following steps:

1) Identify and handle missing values: It is imperative that we identify and handle missing values in our data, failure to do so can lead to inaccurate results and and we might reach faulty conclusions which could be detrimental to our study. As luck would have it, we found no missing data. In the hypothetical unfortunate event that we did find missing data, we would have either deleted the particular row or calculated the mean of the missing feature and assign it to the missing data point. It is worthy to note that the latter method only applies to numerical data.

2) Encoding the categorical data: Categorical data is collection of data that is divided into groups, i.e. Gender is represented by Female or Male. In our data set, we have 12 features which are categorical and we need to assign numerical values to these features due to the fact that Machine Learning algorithms are based on mathematical, making them better suited for numerical values.
In this study, we will be encoding our categorical features using One-Hot-Encoding. With One-Hot-Encoding, we remove the categorical variable and assign a new binary value to the unique categorical value, i.e. for Gender, we split it into two feature Gender_0 and Gender_1, where 1 is assigned for every instance where Female appears for Gender_0 and 0 is assigned to all instances that are Male, the inverse is true.
We used One-Hot-Encoding over Label Encoder due to the fact that Machine Learning algorithms treat the order of numbers as an attribute of significance, meaning that higher numbers hold more value over lower numbers, thus increasing the bias of the algorithm.
Once we have encoded our categorical features in our data set, we increased our features from 16 to 78.

3) Feature scaling: Some Machine Learning algorithms are sensitive to varying degrees of magnitude, units and range. This is where feature scaling is introduced. Feature scaling is a method used to normalize the varying independent features of a data set and it is very crucial that we perform this step as it is detrimental for our Machine Learning algorithms results. You can perform feature scaling in two ways:

- **Normalization:** Normalization is a scaling technique where the data points are shifted and re-scaled in a way that they are bounded in the [0,1] range. The formula for normalization is as follows:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

  where X_{max} and X_{min} represent the maximum and minimum values of the feature respectively.
- **Standardization:** Standardization is a scaling technique where the data points are centered around the mean with a standard deviation of 1. The formula for standardization is as follows:

$$X' = \frac{X - \mu}{\sigma} \tag{2}$$

  where the mean and standard deviation of the feature values is represented by μ and σ respectively.

In this study, we will use Standardization for scaling our features.

4) Splitting the dataset: For every Machine Learning algorithm, we need to split the data into a training and test data sets. Training data set refers to the subset of the original data that is used to train the Machine Learning model and the test set is the subset of the original data that is used to predict the outcomes. Due to our data set being so small, we used a 75:25 training and testing split.

C. Models & Evaluation Metrics

In this section we will provide a high level overview of the Machine Learning (ML) models used in this study. Due to the small size of our data set, we used 10-fold Cross-Validation when training models. The procedure uses a parameter called k that refers to the number of groups that the training data sample is to be split into, and for this study we chose k=10. We will also use accuracy and confusion matrix as evaluation metrics for our models.

1) Gaussian Naïve Bayes: Gaussian Naïve Bayes classifier is a supervised ML algorithm is derived from the Naïve Bayes that follows the Gaussian normal distribution for continuous data variables. The Naive Bayes classifiers are based on the Bayesian Theorem [9], which is used to calculated conditional probability, and the formula for the Bayesian Theorem is given as follows:

$$P(A|B) = \frac{P(A) * P(B|A)}{P(B)} \tag{3}$$

where:

- P(A) = Probability of A occurring
- P(B) = Probability of B occurring
- P(B|A) = Probability of B occurring given A
- P(A|B) = Probability of A occurring given B

For continuous data, Gaussian Naïve Bayes makes an assumption the the continuous variable from each feature is distributed along the Gaussian Distribution. Gaussian Naïve Bayes uses the following equation to calculate the likelihood of data point belonging to a certain feature [1]:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2})} \tag{4}$$

2) Support Vector Machine: Support Vector Machine (SVM) is a Supervised ML algorithm that is mostly used for classification and regression problems. This algorithm plots all data point in the n-dimensional space (n being the number of features we have in our data set). Then it finds a n-dimensional hyper-plane that separates all the features into distinct classes. It does this using the kernel trick which basically transforms low dimensional data points to higher dimensional spaces. For this study to maximise our accuracy for our SVM model we used the Radial Basis Function (rbf) kernel, given by the equation:

$$K(x, x') = exp(-\frac{(||x - x'||^2)}{2\gamma^2}) \tag{5}$$

where x,x' are two points and γ is the variance and hyper-parameter.

3) Random Forest: Random Forest is a Supervised ML algorithm that is mostly used for classification and regression problems. This algorithm creates different training subsets from the given training data (this is known as **Bagging**), then it takes the decision tree with the majority vote for classification problems.

What distinguishes Random Forest from most ML algorithms is the fact that it can handle continuous and discrete data sets, for regression and categorical problems respectively, although it performs better for the latter.

4) Logistic Regression: Logistic regression is a classification model, derived by transforming linear regression cost function using the sigmoid cost function ([10]). The most common logistic regression models are binary, multinomial and ordinal logistic regression. For this study we will focus on the multinomial logistic regression, where there are more than two possible discrete outcomes.

In machine learning, to predict which class a data point belongs to, we set a decision boundary (threshold) and obtain the probability of the the data set falling into the said class using the sigmoid cost function, given by the equation:

$$h\theta(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{6}$$

which returns a probability score in the range of [0,1]. If $h\theta$ is greater or equal to the decision boundary we can deduce that the data point belongs to the feature.

The cost function of a linear regression model is given as:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^i - y^i)^2) \tag{7}$$

to get an equation for Logistic Regression, we make use of the sigmoid cost function to transform the linear regression cost function to get the logistic regression function, and we get:

$$J(\theta) = \frac{1}{m} \sum [y^i log(h\theta(x(i))) + (1 - y^i) log(1 - h\theta(x(i)))] \tag{8}$$

Then we reduce our cost function using the gradient descent.

5) K-Nearest Neighbour: K-Nearest Neighbor is a supervised MaL algorithm used for classification problems. This classifier is commonly based on the Euclidean or Manhattan distance between two points, x = (a,b) which is taken from the test sample and y = (c,d) which is taken from the training sample. For this study we utilized the Manhattan distance, which is given by the equation:

$$distance(x, y) = |a - c| + |b - d| \tag{9}$$

Using this distance metric the KNN algorithm calculates the distance between all other points in the data sets against the one we are trying to classify (e.g point x) and assumes that x belongs to the class of the other point that has the smallest distance.

D. Feature Selection

When trying to identify trends, each feature in our data set represents a patter [11]. In this section, our aim is to choose features that will allows us to deduce patterns in our data and we do this by removing features that have low variance.To obtain our optimal set of features, we used a dimensionality reduction technique called feature selection. Feature selection methods can be categorised by filter, wrapper and embedded methods [12]. In this study we will focus on filter methods, particularly mutual information [3], [13].

1) Mutual Information: Mutual Information is a filter method that calculated the reduction in entropy by calculating the information gain of each independent feature against the dependent feature and select. The mutual information between features X and Y is calculated using the following equation:

$$I(X,Y) = H(X) - H(X|Y) \tag{10}$$

Where I(X,Y) is the mutual information for X and Y, H(X) is the entropy for X and H(X|Y) is conditional entropy for X given Y [14], [15].

Mutual information is always with the [0,1] range, and the larger the value, we see a strong correlation between the two features. If the calculated result is given as zero, then X is independent of Y [15], [16]. After performing information gain on our encoded feature set, we kept 23 features TABLE II and removed 49 features. We achieved the optimal subset with 82% accuracy with a standard deviation of $\sigma = 13\%$ Figure 1

TABLE II
LIST OF FEATURES AFTER FEATURE SELECTION

No.	Feature	Entropy
1	Visited Resource	0.3675
2	Raised Hands	0.3218
3	Student Absent Days: Under 7	0.3003
4	Student Absent Days: Over 7	0.2906
5	Grade ID: G-09	0.1964
6	Grade ID: G-07	0.1522
7	Place of Birth: Lybia	0.1319
8	Grade ID: G-04	0.1255
9	Nationality: Lybia	0.1204
10	Student Absence Days	0.1107
11	Place of Birth: Iraq	0.1025
12	Grade ID: G-07	0.0952
13	Discussion	0.0792
14	Place of Birth: Syria	0.0697
15	Grade ID: G-08	0.0688
16	Relation: Mum	0.0676
17	Parent Answering Survey: Yes	0.0644
18	Section ID: A	0.0581
19	Nationality: Jordan	0.0564
20	Topic: Arabic	0.0562
21	Nationality: Syria	0.0473
22	Place of Birth: Tunis	0.0436
23	Place of Birth: Jordan	0.0434

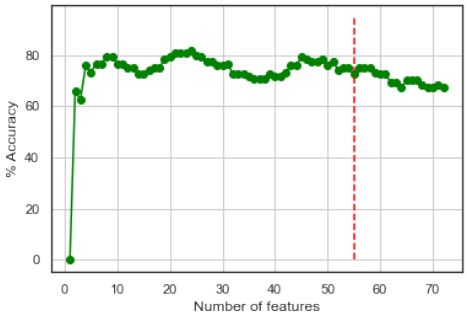

Fig. 1. Accuracy of Mutual Information for the Feature Selection Process

IV. Results And Discussion

In this section we will present the results we got from training out machine learning models after applying mutual information to our feature set. Thereafter we will discuss the performance of each model by using accuracy, recall and precision as our numeric metrics and confusion matrix as our visual metric.

Figures 2 - 6 and TABLE III highlight the performance of the predictive models. Figure 2 demonstrates the confusion matrix for the Gaussian Naïve Bayes model which achieved 50.83% accuracy using 10-fold cross-validation [17], which is the worst classification accuracy achieved compared to the other four models used in this paper this took. With the exclusion of KNN,his model has the slowest build time compared to SVM, Logistic Regression, Gaussian Naïve Bayes and Random Forest.

Figure 3 demonstrates the confusion-matrix for the Logistic Regression model which achieved 74.17% accuracy using 10-fold cross-validation. With the exclusion of the KNN, SVM and Gaussian Naive Bayes, this model has the slowest build time in comparison to Random Forest.

Figure 4 demonstrates the confusion-matrix for the KNN model, which achieved 75% accuracy using 10-fold cross-validation. Furthermore, compared to the other four models used in this paper this model had the fastest build time.

Figure 5 demonstrates the confusion-matrix for the Random Forest classification model, which achieved 78.33% accuracy using 10-fold cross-validation. This model has the slowest build time in comparison to all the other classification models used in this paper.

Figure 6 demonstrates the confusion-matrix for the Support Vector Machine model,which achieved 81.67% accuracy and is the best performing model compared to the other four models used in this paper. Furthermore, compared to the other four models used in this paper this model had the slowest build time.

Given that Support Vector Machine is our best performing, we used it in our feature selection process. TABLE II shows the set of features with variance in descending order [1]. We see that the top 3 features fall under the behavioural features [2], [1].

The top three features are:

- Visited Resources: How frequently did the student utilize the provided resources
- Raised Hands: The amount of times the student has raised their hands in class to ask question or interact with their educator
- Student Absent Days: The number of days the student has been absent from school

This is indicative of how students behavioural attitude towards their academics has a high impact on their performance.

V. Conclusion

Evaluation of our models has demonstrated that pre-processing enhances the performance of our predictive models

TABLE III
PERFORMANCE OF MODELS

Model	Accuracy
Random Forest	78.33%
K-Nearest Neighbour	75.00%
Support Vector Machine	81.67%
Logistic Regression	74.17%
Gaussian Naïve Bayes	50.83%

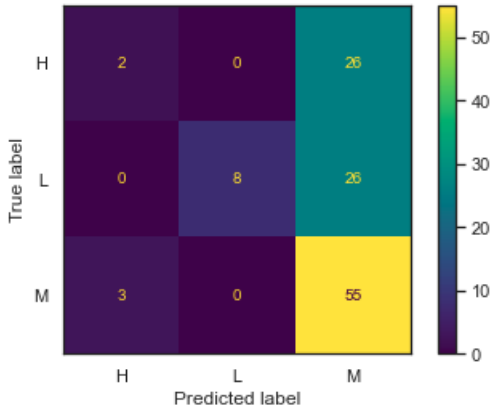

Fig. 2. Confusion Matrix for Gaussian Naïve Bayes Model.

as we have reduced the percentage of our models bias significantly. In this study, we used mutual information gain which has helped us identify the attributes in our data that play a critical role in the students academic performance.

Evaluation results of ranking have shown three features that have a significant impact on the success of the study: Visited Resources, Raised Hands and Student Absent Days. The analysis on the top features offers a conclusion that there is a high correlation between students academic performance and their behavioural attitude.

With this knowledge we can help identify at risk students timeously and provide the needed support to help them improve their academic performance.

ACKNOWLEDGMENT

I would like to acknowledge my supervisors Dr Ritesh Ajoodha and Dr Ashwini Jadhav,for sharing their expertise and valuable guidance. I would like to also thank my mother for her unceasing encouragement, support and attention.

REFERENCES

[1] Eluwumi Buraimoh, Ritesh Ajoodha, and Kershree Padayache. Prediction of student success using student engagement with learning management system. 2021.

[2] Elaf Abu Amrieh, Thair Hamtini, and Ibrahim Aljarah. Mining educational data to predict student's academic performance using ensemble methods. *International Journal of Database Theory and Application*, 9(8):119–136, 2016.

[3] Bruno Trstenjak and Dzěnana Donko. Determining the impact of demographic features in predicting student success in croatia. 2014.

[4] Edin Osmanbegovic and Suljic Mirza. Data mining approach for predicting student performance. *Economic Review: Journal of Economics and Business*, 10(1):3–12, 2012.

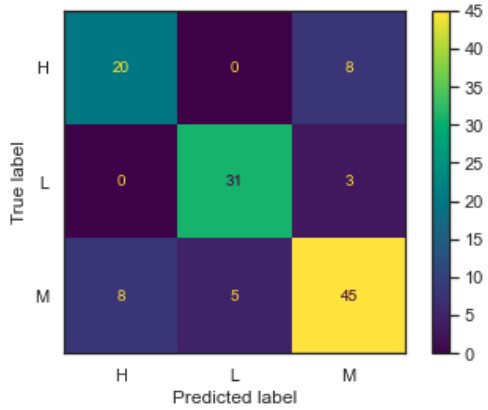

Fig. 3. Confusion Matrix for Logistic Regression Model.

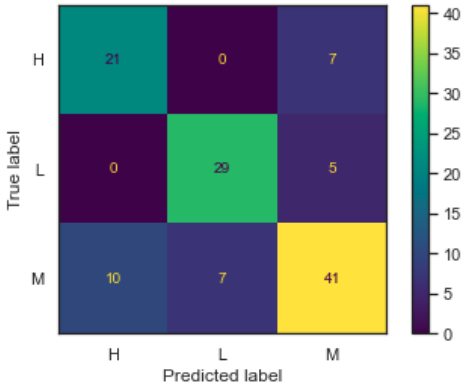

Fig. 4. Confusion Matrix for K-Nearest Neighbour Model.

[5] Dorina Kabakchieva. Predicting student performancee by using data mining methods for classification. *Cybernetics and Information Technologies*, 13(1), 2013.

[6] V. Ramesh, P. Parkavi, and K. Ramar. Predicting student performance: A statistical and data mining approach. *International Journal of Computer Applications*, 63(8), 2013.

[7] Dursin Delen. Predicting student attrition with data mining methods. *J. College Student Retention*, 13(1), 2011.

[8] Dorina Kabakchieva. Predicting student performance by using data mining methods for classification. *Cybernetics and Information Technologies*, 13(1), 2013.

[9] Soner Yildirim. 11 most common machine learning algorithms explained in a nitshell, July 2020.

[10] Rajkumar Buyya, Rodrigo N. Calheiros, and Amir Vaahid Dastjerdi. *Big Data Principles and Paradigms*. Morgan Kaufmann, 2016.

[11] Pablo A. Estévez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada. Normalized mutual information feature selection. *IEEE Transactions On Neural Networks*, 20(2), 2009.

[12] Nuhu Ibrahim, Simon Fong, and Shuzlina Abdul Rahman. Featuree selection methods: Case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science and Technology*, 26(1), 2018.

[13] Ritesh Ajoodha, Ashwini Jadhav, and Shalini Dukhan. Forecasting learner attrition for students' success at a south african university. 2020.

[14] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 4 edition, 2016.

[15] Jason Brownlee. Information gain and mutual information for machine learning, October 2019.

[16] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

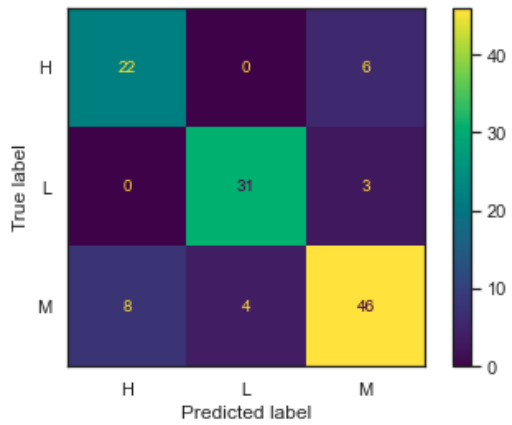

Fig. 5. Confusion Matrix for Random Forest Model.

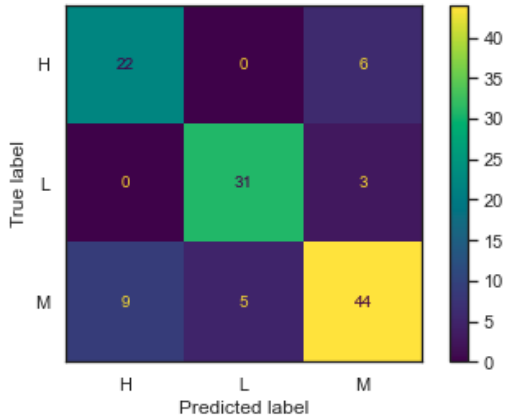

Fig. 6. Confusion Matrix for Support Vector Machine Model.

[17] Ritesh Ajoodha and Ashwini Jadhav. Identifying at-risk undergraduate students using biographical and enrollment observations for mathematical science degrees at a south african university. *ARCTIC Journal: Arctic Institute of North America*, ISSN: 0004-0843, 2019.