Early prediction of at-risk students using OULAD dataset

1.1.1) Abstract:

Online learning platforms, such as MOOCs, VLEs, and LMS, have transformed education by providing access to high-quality resources for millions of people worldwide. However, these platforms face challenges such as student engagement, lack of interest, and self-regulated learning skills. One potential solution to address these challenges is predictive models that can identify at-risk students early in the course to intervene and avoid dropouts.

This project proposes a baseline predictive model using machine learning algorithms to analyze student behavior data, including engagement intensity, assessment scores, and time-dependent variables. Feature engineering was used to identify the most important study variables for predicting at-risk students, and the model was trained and tested on a dataset of student behavior data from an online learning platform. The decision tree model achieved an accuracy of 92.4% and can identify the most important factors for predicting at-risk students.

Instructors can use this information to intervene with appropriate strategies to improve student engagement and study performance. However, further research is necessary to assess the model's effectiveness and scalability across various online learning platforms. It is also important to consider the ethical implications of using predictive models in education to ensure that the model's use does not discriminate against groups of students.

In summary, this project proposes a predictive model to identify at-risk students early in the course, helping instructors intervene to avoid dropouts. The model's accuracy of 92.4% and the identification of engagement intensity, assessment scores, and time-dependent variables as crucial factors make it a useful tool for supporting online learning. However, further research is necessary to assess its effectiveness and scalability, and ethical implications must be considered to ensure fair use.

In conclusion, our project proposes a baseline predictive model that analyzes the problems faced by at-risk students, facilitating instructors for timely intervention to persuade students to increase their study engagements and improve their study performance. Our results showed that engagement intensity, assessment scores, and time-dependent variables are important factors in online learning. The decision tree model provides an accurate and efficient method for identifying at-risk students early in the course, thus avoiding student dropouts. While the model's accuracy is promising, further research is needed to assess its effectiveness and scalability across various online learning platforms.

1.1.2) Keywords from abstract:

- Predictive models
- Feature engineering
- At-risk students
- Engagement intensity
- Assessment scores
- Time-dependent variables
- Decision tree model
- Correlation matrix

1.2.1) Introduction:

The problem statement of "Early prediction of at-risk students" is an important issue in the field of education, particularly for identifying students who may be at risk of dropping out of school or failing a course early in the semester. Early identification of such students can help educators intervene and provide additional support to help them succeed. To tackle this problem, a machine learning model can be developed using historical data on student performance, attendance, and demographic information. The model can be trained on a large dataset of past students who either succeeded or failed in a particular course, using features such as the student's attendance record, grades in previous courses, socio-economic background, and other relevant factors. The goal of the model is to predict which students are likely to be at-risk early in the semester, before they have fallen too far behind in their coursework.

The study variables considered during feature engineering include engagement intensity, assessment scores, and time-dependent variables. The decision tree model identified these variables as important factors for predicting at-risk students. While the proposed model is a step forward in addressing the challenges faced by online learning platforms, there is still a need for ongoing research to improve and adapt the model for different platforms and contexts.

Note: Please refer to the python notebook parallelly to better understand the project deck.

1.2.2) Problem Statement:

The problem statement of "Early prediction of at-risk students" is an important issue in the field of education, particularly for identifying students who may be at risk of dropping out of school or failing a course early in the semester. Early identification of such students can help educators intervene and provide additional support to help them succeed. To tackle this problem, a machine learning model can be developed using historical data on student performance, attendance, and demographic information. The model can be trained on a large dataset of past students who either succeeded or failed in a particular course, using features such as the student's attendance record, grades in previous courses, socio-economic background, and other relevant factors. The goal of the model is to predict which students are likely to be at-risk early in the semester, before they have fallen too far behind in their coursework.

1.2.3) Data Description:

The dataset used in this project belongs to Open University Online Learning Platform. It contains 7 different csv files:

1. Courses.csv:
    - Contains list of all available modules and their presentations. The columns it consists of are:
        - Code_module – code name of module
        - Code_presentation - code name of presentation and is of the format "xxxxJ/B" where 'xxxx' represents the year and the alphabets 'J' and 'B' represents the semester.
        - Length – length of the module-presentation in days.
2. Assessments.csv:
    - Contains information about assessments for each module-presentation. This csv file contains:
        - Code_module- identiofication code of the module.
        - Code_presentation- identification code of the presentation to which the assessment belongs.
        - Id_assessment- identifdication number of assessment.
        - Date – final submission date of assessment calculated as the number of days since the start of the module-presentation.
        - Weight – weight of each asessment. Sum of all assessments is 100.

    - Weightage for exams is 100 and is treated separately for a given module-presentation.

3. vle.csv:

    - Contains information about available materials in Virtual Learning Environment. Columns are:
        - Id_site - identificatioin number of the material
        - Code_module – identification code for module
        - Code_presentation - identification code of presentation
        - Activity_type - role associated with the material

        - Week_from - which from which the module is planning to be used.
        - Week_to - week until the material is planned to be used.

4. StudentInfo.csv:

    - Information about the student with the result. It contains:
        - Code_module
        - code_presentation
        - id_student – unique identification number for the student
        - Gender – student's gender
        - Region – geographical region where a student belongs from
        - Highest_education – highest student education at the time of the start of the module
        - Imd_band - Index of Multiple Depravation band of the place
        - age_band – student's age
        - Num_of_prev_attempts – number of attempts by a student at a module
        - Disability – presence or absence of a disability in a student
        - Final_result – student's final result

5. studentRegistration.csv:

    - Information about the time of registration for module presentation by a student. It has the following attributes:
        - Code_module
        - Code_presentation
        - Id_student
        - Date_registration – the date of student's registration on the module presentation
        - Date_unregistration - the date of student unregistration from the module presentation. Students successful in completing the course have this value set as null. Students who unregistered have withdrawal as the value of the final_result in the studentInfo.csv file.

6. studentAssessment.csv:

- This file contains results of students' assessments and contains the following columns:
  - Id_assessment
  - Id_student
  - date_submitted - date of submission, measured as the number of days since start of the module presentation
  - Is_banked – status flag indicating the assessment result has been transferred from a previous presentation.
  - Score – student's score for that assessment. Ranges from 0-100. A score lower than 40 is interpreted as Fail.

7. studentVle.csv:

- Contains information about each student's interactions with the Vle materials. Attributes contained in this file are:
  - Code_module
  - Code_presentation
  - Id_student
  - Id_site
  - Date-
  - Sum_click – number of times a student interacts with the material in that day.

1.2.3) Literature Survey:

S. No.	Title	Author(s)	Algorithm used and performance achieved	Problem addressed	Conclusion
1	Predicting at-Risk Students at Different Percentages of Course Length for Early Intervention Using Machine	Muhammad Adnan, Asad Habib, Jawad Ashraf, Shafaq Mussadiq	Random Forest (RF) gives the best results with averaged precision = 0.60%, 0.79%, 0.84%, 0.88%, 0.90%, 0.92%, averaged recall =	1.) The merger operation combined Distinction-Pass and Pass classes, as well as Withdrawn-Fail and Fail	The study developed predictive models using machine learning and deep learning algorithms to predict at-risk

	Learning Models		0.59%, 0.79%, 0.84%, 0.88%, 0.90%, 0.91%, averaged F-score = 0.59%, 0.79%, 0.84%, 0.88%, 0.90%, 0.91%, and average accuracy = 0.59%, 0.79%, 0.84%, 0.88%, 0.90%, 0.91% at 0%, 20%, 40%, 60%, 80% and 100%	classes. This was done because they provide similar information. 2.) Feature engineering was applied to improve the performance of predictive models, especially for the at-risk Fail class who require guidance.	students' performance based on demographics, clickstream, and assessment variables. The RF predictive model demonstrated effectiveness in predicting students' performance at different stages of the course length, even with just demographics variables. The study highlights the importance of timely interventions to improve student performance and suggests the need for more in-depth studies to evaluate various online activities and intervention techniques.

2	Predicting Students Success in Blended Learning—Evaluating Different Interactions Inside Learning Management Systems	Luiz Antonio Buschetto Macarini, Cristian Cechinel, Matheus Francisco Batista Machado, Vinicius Faria Culmant Ramos, Roberto Munoz	Thirteen dataset combinations together with five classification algorithms (k-Nearest Neighbor, Multilayer Perceptron, Naive Bayes, AdaBoost and Random Forest) were used in the experiments. It is possible to say that the models achieved performances that can be considered satisfactory (with AUC ROC values of 90% already in the first week)	1.) The use of the SMOTE (Synthetic Minority Over-sampling Technique) technique to balance datasets helps on improving the performance of the models has proved helpful for prediction. 2.) The novelty of this paper's approach is based on the extensive comparison of datasets and classification algorithms, resulting in 65 combinations (13 datasets and 5 classification algorithms).	In conclusion, this study investigated the effectiveness of EDM techniques for early detection of at-risk students and compared different combinations of classifiers and datasets. The results showed that a structured course with a variety of resources and opportunities for student engagement led to better outcomes. Despite limitations in the number of cases, this research contributes to the understanding of how to identify and support at-risk students in introductory programming courses.

3	"Machine Learning Approaches for Student Performance Prediction" using UCI dataset	Shelly Gupta, Jyoti Agarwal	1.) The model classifies students into PASS or FAIL categories using two machine learning algorithms: kNN and Decision Tree. The kNN algorithm stores data and classifies new data points based on similar features, while the Decision Tree creates a tree-like structure to make decisions based on significant attributes. kNN gives 90.75% accuracy, while Decision Tree gives 91.5% accuracy. 2.) The proposed work also uses Logistic Regression algorithm for classification, which predicts categorical dependent variables using independent variables. Logistic Regression outputs a discrete value between 0 and 1. The algorithm gives 85.71% accuracy	1) End-to-end application (frontend- HTML, CSS; backend- flask, pickle) 2) The research analyzed past studies that predicted student achievement using various analytical methodologies but found that relying solely on grade points is insufficient for accurate predictions. The paper suggests incorporating external factors, such as family background, health status, and geographical location, in addition to academic grades for more efficient prediction of student performance.	The study applied KNN classifier and Logistic Regression algorithms on a UCI dataset and found that KNN performed better in terms of accuracy. The proposed model was compared with existing models, showing its efficiency in extracting insights from data and assisting educators in improving student performance. Future research can consider more factors and apply deep learning algorithms for more accurate results in less computation time.

4	Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS	Rianne Conijn; Chris Snijders; Ad Kleingeld; Uwe Matzat	This paper discusses the challenge of creating scalable and versatile models for predicting student performance, rather than proposing a new algorithm. The study examined correlations between 23 predictor variables and final exam grades across multiple courses. Only the between-assessment grade and midterm grade consistently correlated with final exam grade across courses. Other variables had varying degrees of correlation and stability, suggesting the need for personalized prediction models..	The study investigated whether LMS data can predict student performance and explain variance. A multi-level analysis was conducted with crossed-random effects for course and student. LMS data explained part of the variance in final exam grade at the student and course level. However, the amount of explained variance differed across courses. More online sessions, lower standard deviation of time between sessions, and starting early all correlated with a higher grade. Prediction models were not very accurate and were based on complete LMS data at the end of the course.	**1.) Portability of Prediction Models:** Variable sets cannot consistently predict student performance across different courses in a learning management system. The in-between assessment grade was the only variable that correlated significantly with final exam grade in all courses, while other variables showed low correlation or no predictability. **2.) Predicting Student Performance Per Course** Running separate regressions for each course to predict student performance, a weak correlation was found between LMS data and final grades with low accuracy (8-37%). LMS data was not strong enough for precise prediction of early intervention or pass-fail probabilities, possibly due to the lack of relation between LMS activities and the

					final exam in some courses.

5	Predicting student performance: an application of data mining methods with an educational Web-based system	B. Minaei-Bidgoli; D.A. Kashy; G. Kortemeyer; W.F. Punch	This study uses commonly used classifiers such as Quadratic Bayesian, I-nearest neighbor (I-NN), k-nearest neighbor (k-NN), Parzen-window, multilayer perceptron (MLP), and Decision Tree. Preprocessing was performed on the dataset, and the error rates of each classifier were reported. To improve performance, a combination of classifiers was used.	In the case of 3-classes and 9-classes, CART has the best accuracy of about 60% in 3-classes and 43% in 9-Classes. However, considering the combination of non-tree-based classifiers, the CMC has the best performance in all three cases. That is, it achieved 86.8% accuracy in the case of 2-Classes, 71% in the case of 3-Classes) and 51% in the case of 9-Classes.	Four classifiers are used to segregate the students. A combination of multiple classifiers was used that improved the accuracy significantly. An approach using Evolutionary Algorithms tri find association rules and dependency among the groups of problems (Mathematical, Optional Response, Numerical, Java Applet, and so forth) of LON-CAPA homework data sets.
6	A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs	Jie Xu ; Kyeong Ho Moon; Mihaela van der Schaar	1) Novel algorithm based on students' progressive performance stats. Incorporates a bilayered structure comprising a base predictor layer and an ensemble predictor layer. 2) Developed a data-driven course clustering method based on probabilistic matrix factorization. Autonomously, output course clusters based on heterogenous sparse student course grade log/data.	Base prediction layer (lin. Regr./log. Regr./random forest/kNN) combined with a base prediction layer integrating online and offline learning	Novel method proposed given current and past performance. Latent factor model-based course clustering method. Ensemble-based progressive prediction architecture to incorporate students' evolving performance

7	EKT: Exercise-Aware Knowledge Tracing for Student Performance Prediction	Qi Liu; Zhenya Huang; Yu Yin; Enhong Chen	general Exercise-Enhanced Recurrent Neural Network (EERNN) **Performance – RMSE value: 0.42 MAE value: 0.34 ACC value: 0.74 AUC value: 0.74**	The use of attention mechanism proved to improve performance drastically for the neural network model	The paper proposes EERNN and EKT frameworks for predicting student performance, with experiments demonstrating their effectiveness. EERNN uses exercise content and exercising records, while EKT incorporates knowledge concepts of each exercise.
8	Combining University Student Self-Regulated Learning Indicators and Engagement with Online Learning Events to Predict Academic Performance	Abelardo Pardo; Feifei Han; Robert A. Ellis	PCA followed by multiple regression 3 variables, test anxiety (β=−.22, p<.05), Resource (β=.85, p<.01), and MCQ (β=.31, p<.05) significantly contributed to academic performance.	Multiple regression performed in coherence with PCA with different sets of variables where each set of variables gave better performance for one target variable more than the others	The study used self-reported and observed data to analyze a blended learning course and showed a linear model explaining 32% of academic performance variance. The results suggest further exploration of combining data sources for teaching and learning improvement.
9	Predicting Student Performance Using Personalized Analytics	Asmaa Elbadrawy; Agoritsa Polyzou; Zhiyun Ren; Mackenzie Sweeney	Personalized multi-regression and matrix factorization (MF) approach based on recommender systems	Combines LMS and MOOC data for more accurate prediction	Recommender systems accurately predict student performance with low error rates using PLMR and advanced MF techniques, which incorporate historical and additional data.

10	Modeling engagement of programming students using unsupervised machine learning technique	Hua Fwa, Emeritus Professor Lindsay Marshall	Lin. Regression, log. Regression, random forest, kNN, EPP> Ensemble-based progressive prediction showed the best result having the lowest mean square error	Tracking and predicting students' performances using ML techniques	used HMM to infer engagement states of students from their actions and sensors. Three states were identified, engaged, starting out, and disengaged, and interventions were suggested to shift students to more enduring engagement.
11	Machine Learning Based Student Grade Prediction: A Case Study	Zafar Iqbal, Junaid Qadir, Adnan Noor Mian, and Faisal Kamiran*	Collaborative Filtering (CF), Matrix factorization (MF), Restricted Boltzmann Machines (RBM) techniques,RBBM proved to be the best with rmse = 0.3, mse = 0.09, mae = 0.23	Model programming students using unsupervised ML techniques	RBM technique predicts student performance in courses, aiding early warning and counseling for students to improve their knowledge in certain areas. The technique also helps instructors identify and intervene with weak students, increasing retention rates.
12	What Time is It? Student Modeling Needs to Know	Ye Mao; Samiha Marwan; Thomas W. Price; Tiffany Barnes	Bayesian Knowledge Tracing (BKT), Intervention-BKT(IBKT), LSTM, LSTM and LSTM+SK achieved the highest accuracy of 74%	Implementation of time-aware LSTM(T-LSTM) as the time elapsed between successive elements in a student's trajectory can vary from seconds to days.	T-LSTM is effective in predicting student success in open-ended programming but not in well-defined domains. More research is needed for generalization and improved models.

13	Systematic Literature Review on Machine Learning and Student Performance Prediction: Critical Gaps and Possible Remedies	Boran Sekeroglu, Rahib Abiyev, Ahmet Ilhan, Murat Arslan and John Bush Idoko	SVR, LSTM, SVM, Gradient Boosting Classifier (GBC), ANN. ANN with the highest accuracy of 74%	Students' performance classification and prediction using ML techniques.	This study reviews student performance prediction studies in artificial intelligence in education and suggests using specific evaluation metrics and validation techniques. Future studies are expected to focus on deep learning with expanding data and computer technologies.
14	Prediction of students performance using Educational Data Mining	Tismy Devasia;Vinushree T P;Vinayak Hegde	The proposed system is a web-based application which makes use of the Naive Bayesian mining technique for the extraction of useful information	Large dataset processed through data mining. Mechanism works in large datasets where the student performance in the end semester examination is evaluated.	This paper uses Naive Bayes classification to predict students' performance based on their previous academic data. The study aims to identify students who need extra attention and improve overall performance. Future work includes using data processing techniques to improve accuracy and efficiency with a larger data set.

15	Predicting Instructor Performance Using Data Mining Techniques in Higher Education	Mustafa Agaoglu	Proposed model – C5(variant of decision tree) with accuracy 92.3%	All models trained gave a performance greater than 90% reinforcing the importance of data mining.	study applies data mining techniques to course evaluation questionnaires to identify variables that differentiate satisfactory and unsatisfactory instructor performances. The study shows that data mining techniques can be effective in higher education and contribute to improvements in measurement instruments.
16	A Robust Machine Learning Technique to Predict Low-performing Students	SOOHYUN NAM LIAO, DANIEL ZINGARO, KEVIN THAI, CHRISTINE ALVARADO, WILLIAM G. GRISWOLD, and LEO PORTER	support vector machines (SVMs) with the radial basis function kernel to train one prediction model. On average, the AUC and 95% confidence interval of the courses are 0.70 and 0.63–0.76	Model uses only student clicker responses from lectures. Relatively lightweight	This work proposes a support vector machine binary classification method to identify at-risk students early in a course. The approach can predict students in different terms, courses, and institutions, and requires only data collected during teaching.

17	"Turn on, Tune in, Drop out": Anticipating student dropouts in Massive Open Online Courses	Diyi Yang, Tanmay Sinha, David Adamson, Carolyn Penstein Rose	Traditional machine learning models, ML models don't generalize well	Incorporation of "social network" as feature	We aim to understand how bonds form in discussion threads and develop models to predict subcommunity formation and participation. We use mixed membership social network partitioning models and text mining techniques to analyze community structure and aim to support healthy engagement in MOOCs.
18	CLMS-Net: dropout prediction in MOOCs with deep learning.	Nannan Wu	CLMS-Net. Combination of CNN, LSTM, SVM. Accuracy – 91.55%	development of a deep learning model called CLMS-Net, which combines convolutional, long short-term memory (LSTM), and multi-layer perceptron (MLP) layers to predict student dropout in MOOCs.	The paper proposes CLMS-Net, a deep learning-based method that predicts MOOC dropouts using weekly course features. CLMS-Net outperforms other models and helps instructors take proactive measures to prevent dropouts and improve learning outcomes.

19	From Lab to Production: Lessons Learnt and Real-Life Challenges of an Early Student-Dropout Prevention System	Alvaro Ortigosa; Rosa M. Carro; Javier Bravo-Agapito	C5.0 algorithm with more than 95% accuracy	This research paper led to the creation of SPA (Sistema de Predicciæn de Abandono, dropout prediction system in Spanish) which has been in use since 2017	The article discusses how a Spanish distance university uses predictive models to prevent student dropout. Challenges include cost effectiveness, organizational changes, model explainability, legal regulations, and technical adaptation. Future work includes evaluating real-world performance and improving retention strategies.
20	An early warning system to identify and intervene online dropout learners	figueroa-Casas, J. and sancho-vinuesa			
21	Dropout early warning systems for high school students using machine learning	Jae Young Chung, Sunbok Lee	Random Forest (RF) with 95% accuracy	Students' binary classification	The study explores using machine learning to predict student dropouts. The random forests model showed excellent performance. The results demonstrate the benefits of using machine learning with big data in education. The predictive model can be integrated into NEIS to evolve the dropout early warning system in real-time.

22	The application for gaussian mixture models for the identification of At-Risk learners in Massive Online Open Courses	Raghad AL-Shabandar, Andy Laws, Thar Baker	Gradient Boosting Model with 95% accuracy	Identification of at-risk students with intensive earlier intervention in online courses	Study predicts at-risk learners in MOOCs. VLE activities measured by sessions and clicks. Mixture model identifies at-risk students. Latent engagement affects persistence. Difficulty affects engagement.
23	Student Academic Performance Monitoring and Evaluation Using Data Mining Techniques	Emmanuel N. Ogor	C5 algorithm with an accuracy of 95%.	The deployment of the prototyped solution integrates measuring, 'recycling' and reporting procedures in the new system to optimize prediction accuracy.	DMT effectively monitored student academic performance with 94% success and improved rule quality with fine-tuning of derived variables. The system's reporting tools compared changes over time and identified systematic structures to improve performance monitoring. OLAP with dynamic reporting is recommended for large student databases in Oracle or MS SQL Server.

24	A Comparison of Regression Models for Prediction of Graduate Admissions	Mohan S Acharya, Asfia Armaan, Aneeta S Antony	Linear regression (among other models) performs the best (MSE-0.005, RMSE-0.07)	Research paper includes parameters that are all relevant for graduate admissions	Linear Regression was the best-performing model among the four evaluated, followed by Random Forest. This is because the dataset's features have linear dependencies, where higher test scores and GPA increase admission chances. However, the Linear model was somewhat influenced by a few outliers.
25	Predicting Students' GPA and Developing Intervention Strategies Based on Self-Regulatory Learning Behaviors	AMIN ZOLLANVARI, REFIK CAGLAR KIZILIRMAK, YAU HEE KHO2 AND DANIEL HERNÁNDEZ-TORRANO	Maximum-Weight Dependent Trees (MWDT) has an accuracy of 65.85% with a sensitivity of 63.9% and a specificity of 67.4%.	GPA prediction to identify students requiring early intervention	We developed a model to predict GPA based on students' self-regulatory behaviors, achieving 65.85% accuracy. Our aim is to help struggling students improve their performance with intervention strategies based on self-regulation. Further research is needed to improve accuracy by including additional variables and combining them with prior performance.

1.2.3) Flow of paper:

- Section 1 explains the problem statement, background, metadata and other important inferences from dataset (skewness, sampling, etc.)
- Section 2 talks about the inferences of the baseline model’s performance and justify its performance
- Section 3 contains future scope ideas and workflow for the next phase.

1.3) Why I decided to pursue this project:

I took up this problem statement as a research paper because I am interested in exploring the potential of machine learning algorithms in improving student engagement and performance in online learning environments. The rise of online learning platforms such as MOOCs and VLEs has created a paradigm shift in education, providing access to high-quality educational resources to millions of people worldwide. However, these platforms also face significant challenges, such as student disengagement and dropout rates.

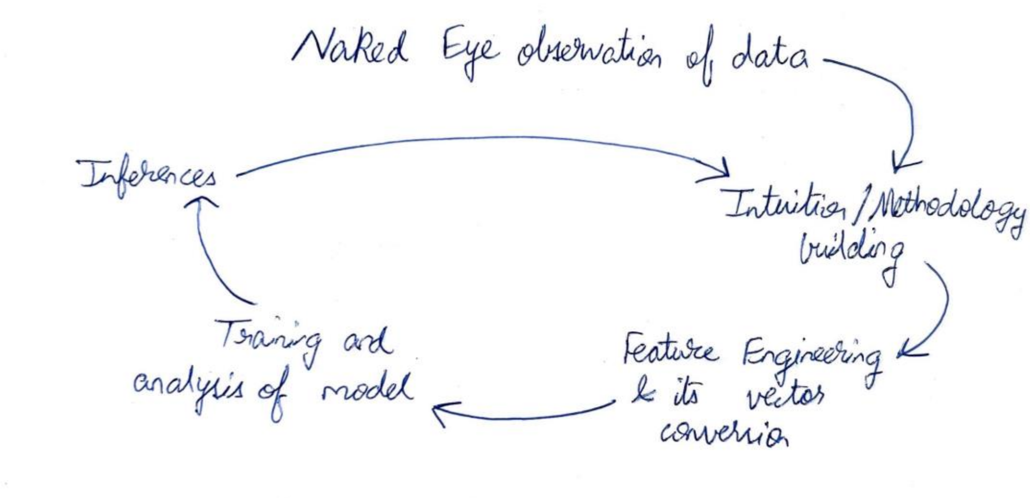
By developing a predictive model that can identify at-risk students early in the course, we can intervene with appropriate strategies to increase engagement and improve student performance. I believe that this has the potential to significantly reduce student dropout rates and improve overall student outcomes. Additionally, machine learning algorithms can provide valuable insights into the factors that contribute to student engagement and performance, which can inform the development of more personalized and effective learning experiences for students. Overall, I am excited about the potential of this research to contribute to the ongoing efforts to improve online learning platforms and provide better educational opportunities for all.

1.4) Background:

- Multi-class classification is a type of machine learning task where the goal is to assign a given input to one of several possible categories or classes. In other words, it involves predicting a single output variable with more than two possible values.

  Note: Multi label classification is not to be confused with multi class classification, a mistake I made in the developing process. In Multiclass the classes are mutually exclusive, while in Multilabel each label represents a different class. Multiclass classification means a classification problem where the task is to classify between more than two classes. Multilabel classification means a classification problem where we get multiple labels as output.

1.5) Workflow:



- The workflow for a typical machine learning project involves several essential steps. Firstly, the problem statement must be clearly defined, along with the goals of the project.
- Secondly, the data must be explored and analyzed, without the use of any tools or structures. This stage involves building intuition and deriving features

through manipulation of different attributes, which may not be immediately apparent in the dataset.

- The third step, feature engineering, requires converting intuition into mathematical vectors that can be used as features for the machine learning model. This involves using techniques such as one-hot encoding, feature scaling, and dimensionality reduction, to transform the data into a format that can be understood by the machine learning algorithms.
- Once the features have been engineered, the next step is to select the most appropriate machine learning model(s) for the problem statement. This requires a meticulous analysis of the different models available, including their strengths and weaknesses, to identify the one(s) that will perform best for the given task.
- Finally, after the model has been selected and trained, it is important to evaluate its performance using various metrics. This enables us to assess the model's strengths and weaknesses, identify areas for improvement, and make inferences based on the results.
- In summary, the workflow for a machine learning project involves a series of critical steps, including problem formulation, data exploration, feature engineering, model selection, and performance evaluation. By following this process rigorously and making informed decisions at each stage, we can create effective machine learning models that provide valuable insights and solutions to complex problems.

1.6) Metadata:

- 28785 registered students
- 7 selected courses (AAA, BBB, CCC, DDD, EEE, FFF, GGG)
- 10655280 total registered presentations
- 206 presentations

- **Files:**
  - courses.csv
  - assessments.csv

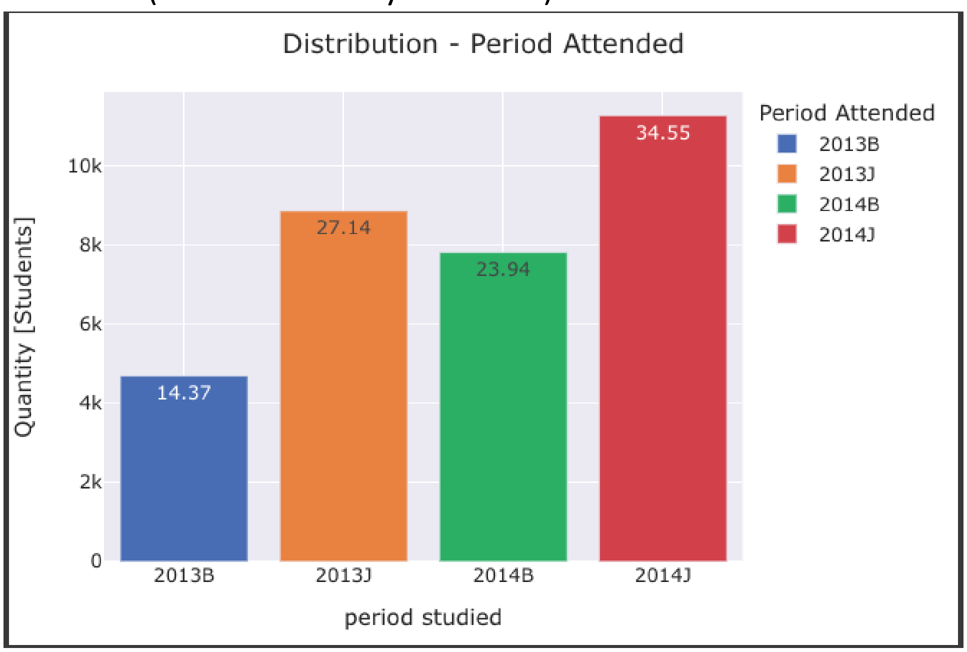
- vle.csv
- studentInfo.csv
- studentRegistration.csv
- studentAssessment.csv
- StudentVle.csv

1.7) Data preprocessing:

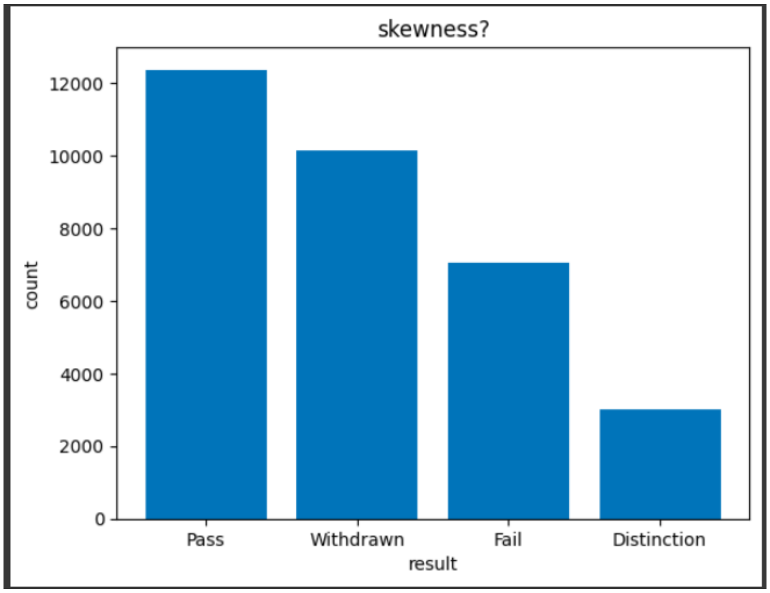
To enhance the efficiency of our predictive models, a decision was made to remove or replace all missing variables (null values) for each attribute in the dataset. However, this step was executed only after the combination of different dataframes to create a comprehensive feature set for training the models. Initially, during the preprocessing stage of the project, this approach was not taken as it would have resulted in the loss of instances that contained other important non-zero attributes.

1.7) Data Sampling/graphs and other info:

1. It seems that around 34.5% of the students are registered for the 2014J semester (the most for any semester)

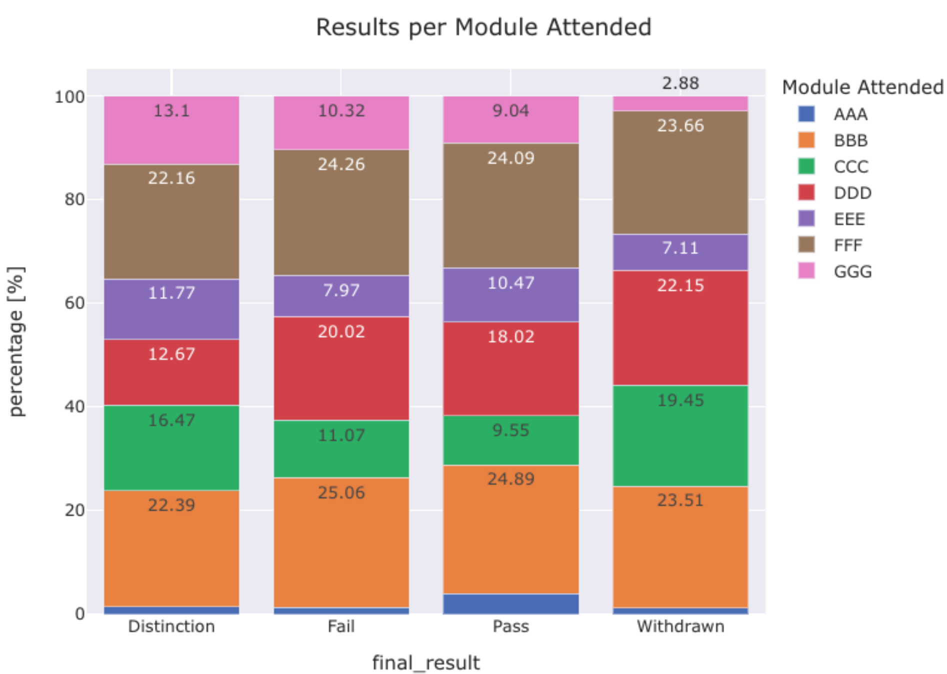


2. This data has been recorded in real time which is why there is considerable skewness in terms of the output variable (>12000 for pass and around 10000 for withdrawn)

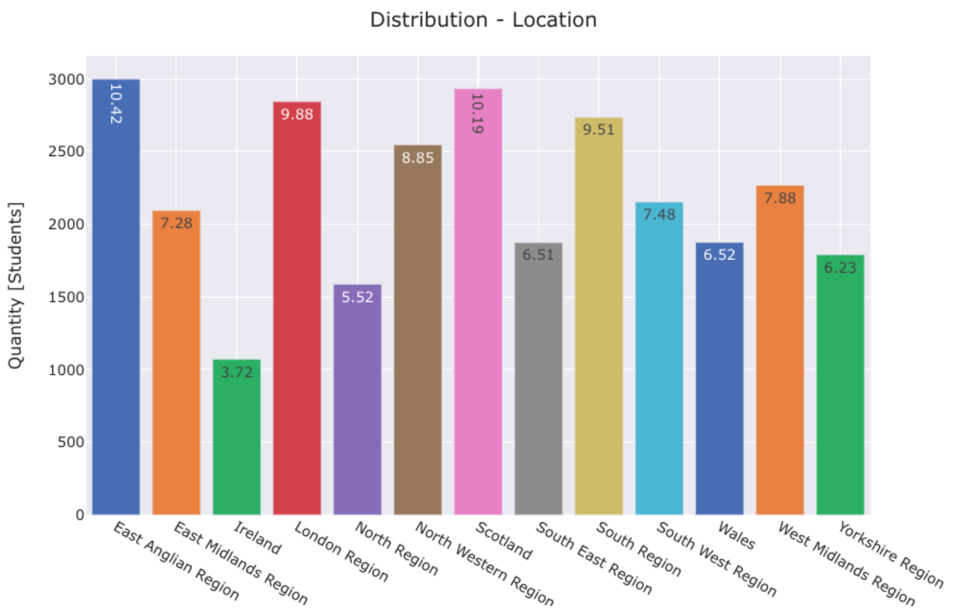


3. Results per module attended

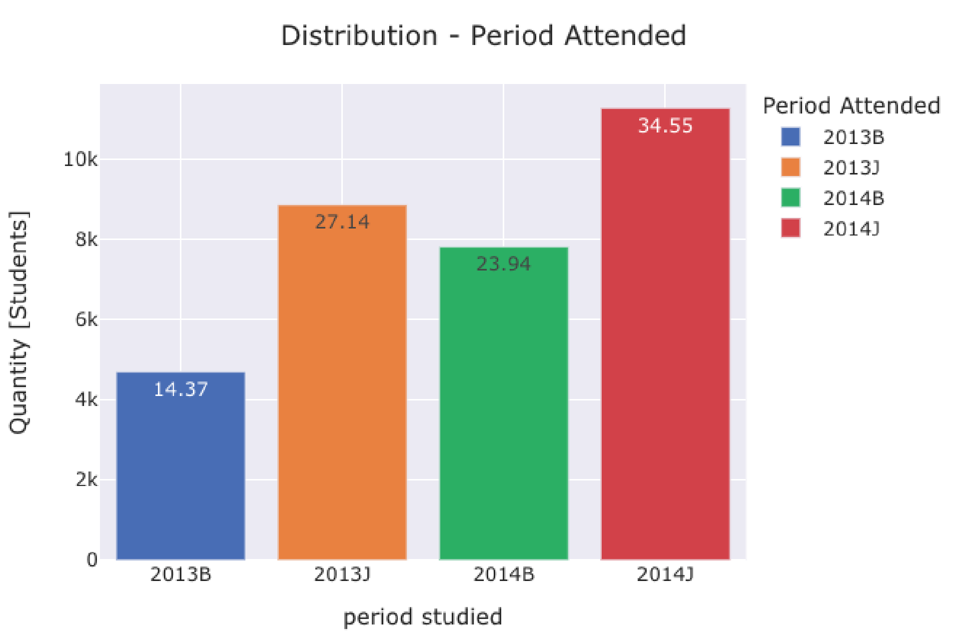
   Highest instances of modules for every result are for Modules FFF and BBB (they also have the highest number of registered students)



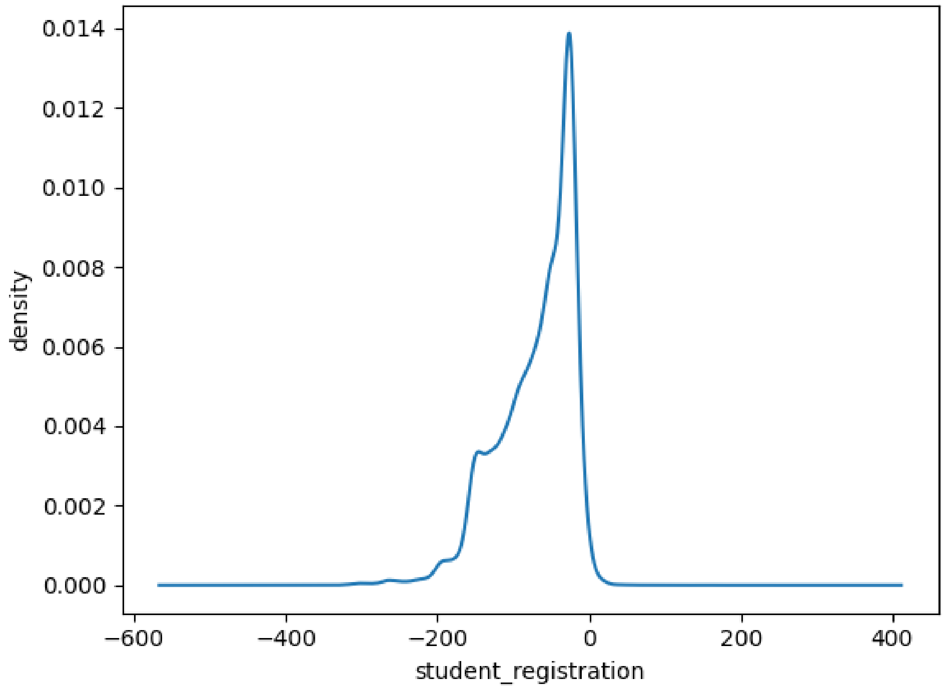
4. Distribution – Location



5. Distribution – Period attended



6. Density v Student registration date



Maximum number of students register just a few days before the start of the module presentation

Note: It should be noted that the date of a student's registration is the number of days measured relative to the start of the module presentation.

7. Imd band v target variable:

Plotting a graph for this feature seemed very counter intuitive as it was too extensive with possible inferences being buried under text. Rather, a dataframe of the graph's coordinates was made in the python notebook and the following inferences were made:

- Around 60% students that failed belong to the imd_band 0-60%(collective of 6 imd groups)
- Around 50% students that withdrew belong to the imd_band 0-50%(collective of 5 imd groups)

8. Sum_clicks v target variable:

Number of students with clicks in the range of 0-2000 are the most for every target output suggesting a pattern.

result

10. Code_module v target variable:

- 65% of students who studied module AAA passed
- 38% of students who studied module BBB passed and 30% withdrew
- 44% of students who studied module CCC withdrew
- 35% of students who studied module DDD passed and 35% withdrew
- 44% of students who studied module EEE passed
- 38% of students who studied module FFF passed
- 44% of students who studied module GGG passed

1.7) Feature Engineering:

To recognize the different attributes across the 7 csv files and their relevance in making our predictive models, we followed a workflow that involved:

- Building an intuition from the ground up to understand what attributes (keeping in mind the number of null values it has) would contribute most to increasing performance.
- Combining different data frames to get feature vectors.
- Label encoding of categoric attributes.
- Checking correlation of every attribute with one another and removing ones which are closely related to others. This is done to reduce the size of the instance to avoid overfitting.

We came down to 4 different feature sets. They are:

1. Feature set used in iteration **1** of the python notebook - ['gender_num','code_module_x_num','code_presentation_x_num','region_num','highest_education_num','imd_band_num','age_band_num','disability_num','num_of_prev_attempts','studied_credits', 'sum_click']

2. Feature set used in iteration **2** of the python notebook - ['sum_click']

3. Feature set used in iteration **3** of the python notebook - ['code_module_x_num','code_presentation_x_num','region_num','highest_education_num','imd_band_num','age_band_num','disability_num','num_of_prev_attempts','studied_credits', 'gender_x', 'date_registration','date_unregistration']

4. Feature set used in iteration **4** of the python notebook - ['code_module_x_num','code_presentation_x_num','region_num','highest_education_num','imd_band_num','age_band_num','disability_num','num_of_prev_attempts','studied_credits', 'gender_x', 'date_registration','date_unregistration', 'sum_click']

2) Evaluation Criteria:

a. Accuracy:

Accuracy in multi-class classification is a performance metric that measures the overall correctness of a classifier's predictions across multiple classes or categories. It is defined as the ratio of correctly classified instances to the total number of instances in the dataset.

To calculate accuracy in multi-class classification, you need to determine the number of instances that were correctly classified for all classes and divide it by the total number of instances:

Accuracy = (Number of correctly classified instances) / (Total number of instances)

b. Precision:

precision is a performance metric that measures the ability of a classifier to accurately identify positive instances for a specific class. It quantifies the proportion of instances that were truly positive (correctly classified) out of all instances predicted as positive for that class.

To calculate precision for a specific class in multi-class classification, you need to determine the number of instances that were correctly classified as positive for that class and divide it by the total number of instances predicted as positive for that class:

Precision = (Number of true positives for a class) / (Number of instances predicted as positive for that class)

c. Recall:

recall (also known as sensitivity or true positive rate) is a performance metric that measures the ability of a classifier to identify all positive instances for a specific class. It quantifies the proportion of instances that were correctly classified as positive for that class out of all instances that actually belong to that class.

To calculate recall for a specific class in multi-class classification, you need to determine the number of instances that were correctly classified as positive for that class and divide it by the total number of instances that truly belong to that class:

Recall = (Number of true positives for a class) / (Number of instances that belong to that class)

d. F1-score:

In multi-class classification, the F1 score is a performance metric that combines precision and recall to provide a balanced evaluation of a classifier's performance. It is the harmonic mean of precision and recall and gives equal importance to both metrics.

The F1 score for a specific class in multi-class classification is calculated as follows:

F1 score = 2 * (Precision * Recall) / (Precision + Recall)

The F1 score ranges between 0 and 1, where a value of 1 represents perfect precision and recall, and 0 represents poor performance.

e. Backward Elimination:

A feature selection method that starts with a model trained on all the features and iteratively removes the least significant feature. It involves fitting the model, evaluating the significance of each feature (e.g., using p-values), and removing the least significant feature at each iteration.

Note: For our use case, recall is prioritized over precision. This is because a high recall ensures minimization of false negatives, ensuring that the classifier does not misclassify target variables such as ‘Fail’ and ‘Withdrawn’ as other target variables.

3) Experimental Results and Discussion:

For every iteration, we implemented two models:

- Decision tree - a machine learning algorithm that is commonly used for both classification and regression tasks. It is a supervised learning algorithm that builds a tree-like model of decisions and their possible consequences. The tree structure consists of internal nodes representing features, branches representing decision rules, and leaf nodes representing the predicted outcomes or values.

- Naïve Bayes - a probabilistic classification algorithm based on Bayes' theorem. It assumes that the features are conditionally independent of each other given the class label. Despite its naive assumption, the model has been found to perform well in many real-world applications and is particularly effective when dealing with high-dimensional data.

  The Naive Bayes model calculates the probability of a particular class label given the observed features using Bayes' theorem:

  - $P(y|x_1, x_2, ..., xn) = (P(x_1, x_2, ..., xn|y) * P(y)) / P(x_1, x_2, ..., xn)$
  - Where:
  - $P(y|x_1, x_2, ..., xn)$ is the posterior probability of class y given the observed features $x_1, x_2, ..., xn$.
  - $P(x_1, x_2, ..., xn|y)$ is the likelihood of observing the features $x_1, x_2, ..., xn$ given the class y.
  - P(y) is the prior probability of class y.
  - $P(x_1, x_2, ..., xn)$ is the probability of observing the features $x_1, x_2, ..., xn$.

A) Iteration 1:

Using feature set - ['gender_num', 'code_module_x_num','code_presentation_x_num','region_num',' highest_education_num','imd_band_num','age_band_num','disability_num','num_of_prev_attempts','studied_credits', 'sum_click']

- Decision Tree:

	precision	recall	f1-score	support
Distinction	0.88	0.92	0.90	484951
Fail	0.93	0.84	0.88	317788
Pass	0.93	0.96	0.94	1432865
Withdrawn	0.94	0.87	0.90	365711
accuracy			0.92	2601315
macro avg	0.92	0.90	0.91	2601315
weighted avg	0.92	0.92	0.92	2601315

- Naïve Bayes:

	precision	recall	f1-score	support
Distinction	0.31	0.20	0.24	484951
Fail	0.25	0.08	0.12	317788
Pass	0.57	0.77	0.66	1432865
Withdrawn	0.25	0.18	0.21	365711
accuracy			0.50	2601315
macro avg	0.35	0.31	0.31	2601315
weighted avg	0.44	0.50	0.45	2601315

- Backward Elimination:

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.2046	0.001	931.247	0.000	1.202	1.207
gender_num	-0.0011	0.001	-2.112	0.035	-0.002	-8.24e-05
code_module_x_num	0.0086	0.000	59.199	0.000	0.008	0.009
code_presentation_x_num	0.0065	0.000	25.507	0.000	0.006	0.007
region_num	-0.0014	6.59e-05	-21.287	0.000	-0.002	-0.001
highest_education_num	0.0915	0.000	276.030	0.000	0.091	0.092
imd_band_num	0.0157	8.61e-05	182.691	0.000	0.016	0.016
age_band_num	0.0533	0.001	104.685	0.000	0.052	0.054
disability_num	0.1053	0.001	115.985	0.000	0.104	0.107
num_of_prev_attempts	0.0735	0.001	119.386	0.000	0.072	0.075
studied_credits	0.0018	7.11e-06	248.116	0.000	0.002	0.002
sum_click	-0.0008	2.88e-05	-28.416	0.000	-0.001	-0.001

Omnibus:	1155958.296	Durbin-Watson:	0.004
Prob(Omnibus):	0.000	Jarque-Bera (JB):	967982.686
Skew:	-0.587	Prob(JB):	0.00
Kurtosis:	2.361	Cond. No.	454.

B) Iteration 2:

Using feature set - ['sum_click']

- Decision tree:

	precision	recall	f1-score	support
Distinction	0.00	0.00	0.00	484951
Fail	0.00	0.00	0.00	317788
Pass	0.55	1.00	0.71	1432865
Withdrawn	0.00	0.00	0.00	365711
accuracy			0.55	2601315
macro avg	0.14	0.25	0.18	2601315
weighted avg	0.30	0.55	0.39	2601315

- Naïve Bayes:

	precision	recall	f1-score	support
Distinction	0.00	0.00	0.00	484951
Fail	0.00	0.00	0.00	317788
Pass	0.55	1.00	0.71	1432865
Withdrawn	0.00	0.00	0.00	365711
accuracy			0.55	2601315
macro avg	0.14	0.25	0.18	2601315
weighted avg	0.30	0.55	0.39	2601315

- Backward Elimination:

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.6494	0.000	5851.844	0.000	1.649	1.650
sum_click	-0.0009	2.91e-05	-31.208	0.000	-0.001	-0.001

Omnibus:	1071375.871	Durbin-Watson:	0.004
Prob(Omnibus):	0.000	Jarque-Bera (JB):	950136.864
Skew:	-0.590	Prob(JB):	0.00
Kurtosis:	2.401	Cond. No.	10.5

C) Iteration 3:

Using feature set - ['code_module_x_num','code_presentation_x_num','region_num','highest_education_num','imd_band_num','age_band_num','disability_num','num_of_prev_attempts','studied_credits', 'gender_x', 'date_registration','date_unregistration']

- Decision tree:

	precision	recall	f1-score	support
Distinction	1.00	1.00	1.00	24139
Fail	1.00	1.00	1.00	46925
Pass	1.00	1.00	1.00	125819
Withdrawn	1.00	1.00	1.00	475236
accuracy			1.00	672119
macro avg	1.00	1.00	1.00	672119
weighted avg	1.00	1.00	1.00	672119

→ Cross Validation scores for 30 iterations to check for overfitting:

array([0.67585253, 0.67459382, 0.64398322, 0.68863596, 0.73609177,
0.60985538, 0.71672023, 0.69658989, 0.69750937, 0.71755044,
0.64703624, 0.69627745, 0.73780575, 0.75596322, 0.74179611,
0.68996608, 0.7247188 , 0.62545081, 0.58930548, 0.68019996,
0.62507588, 0.58115515, 0.64327799, 0.54647872, 0.6145029 ,
0.59908587, 0.54701435, 0.63275873, 0.64047171, 0.58184772])

As the cross-validation scores dip significantly as compared to train and test accuracy, we can conclude that the decision tree model for this dataset is overfitting

- Naïve Bayes:

click to expand	precision	recall	f1-score	support
Distinction	0.09	0.03	0.04	24139
Fail	0.19	0.12	0.15	46925
Pass	0.40	0.05	0.09	125819
Withdrawn	0.72	0.94	0.82	475236
accuracy			0.68	672119
macro avg	0.35	0.29	0.28	672119
weighted avg	0.60	0.68	0.61	672119

- Backward Elimination:

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.4194	0.002	1025.364	0.000	2.415	2.424
code_module_x_num	-0.0077	0.000	-27.419	0.000	-0.008	-0.007
code_presentation_x_num	0.0411	0.000	114.536	0.000	0.040	0.042
region_num	-0.0046	0.000	-42.814	0.000	-0.005	-0.004
highest_education_num	0.0093	0.000	19.617	0.000	0.008	0.010
imd_band_num	0.0134	0.000	98.061	0.000	0.013	0.014
age_band_num	-0.0368	0.001	-43.947	0.000	-0.038	-0.035
disability_num	-0.0076	0.001	-6.552	0.000	-0.010	-0.005
num_of_prev_attempts	-0.2552	0.001	-381.105	0.000	-0.257	-0.254
studied_credits	0.0019	9.56e-06	202.704	0.000	0.002	0.002
gender_x	-0.0352	0.001	-37.794	0.000	-0.037	-0.033
date_registration	0.0003	7.29e-06	40.948	0.000	0.000	0.000
date_unregistration	0.0008	5.05e-06	167.819	0.000	0.001	0.001
sum_click	-0.0007	5.4e-05	-13.196	0.000	-0.001	-0.001

Omnibus:	1071498.467	Durbin-Watson:	0.004
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2832375.394
Skew:	-1.754	Prob(JB):	0.00
Kurtosis:	5.814	Cond. No.	963.

D) Iteration 4:

Using feature set -

['code_module_x_num','code_presentation_x_num','region_num','highest_education_num','imd_band_num','age_band_num','disability_num','num_of_prev_attem pts','studied_credits', 'gender_x', 'date_registration','date_unregistration', 'sum_click']

- Decision tree:

	precision	recall	f1-score	support
Distinction	0.44	0.46	0.45	6055
Fail	0.35	0.36	0.35	6720
Pass	0.68	0.67	0.67	23372
Withdrawn	0.39	0.39	0.39	5272
accuracy			0.55	41419
macro avg	0.46	0.47	0.47	41419
weighted avg	0.55	0.55	0.55	41419

- Naïve Bayes:

	precision	recall	f1-score	support
Distinction	0.38	0.26	0.31	6055
Fail	0.38	0.17	0.23	6720
Pass	0.61	0.81	0.70	23372
Withdrawn	0.32	0.18	0.23	5272
accuracy			0.55	41419
macro avg	0.42	0.36	0.37	41419
weighted avg	0.50	0.55	0.51	41419

- Backward Elimination:

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.0867	0.013	157.214	0.000	2.061	2.113
gender_num	-0.0340	0.004	-8.884	0.000	-0.041	-0.026
code_module_x_num	0.0079	0.001	6.723	0.000	0.006	0.010
highest_education_num	0.0509	0.002	20.556	0.000	0.046	0.056
imd_band_num	0.0064	0.001	10.134	0.000	0.005	0.008
age_band_num	0.0328	0.004	8.333	0.000	0.025	0.040
disability_num	0.0611	0.007	9.343	0.000	0.048	0.074
studied_credits	0.0016	5.1e-05	30.926	0.000	0.001	0.002
date_submitted	-0.0009	2.59e-05	-35.561	0.000	-0.001	-0.001
score	-0.0078	9.91e-05	-78.836	0.000	-0.008	-0.008

Omnibus:	16096.996	Durbin-Watson:	0.298
Prob(Omnibus):	0.000	Jarque-Bera (JB):	14695.183
Skew:	-0.586	Prob(JB):	0.00
Kurtosis:	2.427	Cond. No.	1.20e+03

E) Artificial Neural Network (ANN) implementation:

4) Conclusion:

In this particular study, we proposed different predictive models trained on multiple machine learning (ML) and deep learning (DL) algorithms. These models aimed to predict students' performance based on different sets of variables, including demographics, demographics combined with clickstream data, and demographics combined with clickstream and assessment data. Among the models tested, the random forest (RF) predictive model demonstrated the highest performance scores.

The selected RF predictive model can be a valuable tool for forecasting students' performance throughout the course. Its implementation enables instructors to identify at-risk students and intervene promptly to enhance their study performance. By utilizing the predictive model, instructors can make timely interventions and motivate struggling students to improve their academic outcomes.

Among the variables considered, clickstream and assessment data proved to have the most substantial impact on the final predictions of student performance. Clickstream data captures students' online behavior, such as their interactions with learning platforms, time allocation to different activities, and resource utilization. Assessment data, on the other hand, encompasses students' performance in quizzes, assignments, and exams. By incorporating these variables into the predictive model, instructors can obtain more accurate assessments of students' performance and make more informed interventions.

Overall, the proposed RF predictive model, combined with the inclusion of clickstream and assessment variables, offers instructors valuable insights and support to improve students' study behaviors, intervene effectively, and ultimately enhance students' overall performance and course retention rates.

The proposed model is a decision tree fitted with the first feature set(i.e. 1st iteration) with an accuracy of 92% and recall as 92, 84, 96, 87 for the classes “Distinction”, “Fail”, “Pass”, “Withdrawn”.

5) References:

- M. Adnan et al., "Predicting At-Risk Students at Different Percentages of Course Length for Early Intervention Using Machine Learning Models," in IEEE Access, vol. 9, pp. 7519-7539, 2021, doi: 10.1109/ACCESS.2021.3049446.

- N. Kondo, M. Okubo and T. Hatanaka, "Early Detection of At-Risk Students Using Machine Learning Based on LMS Log Data," 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Hamamatsu, Japan, 2017, pp. 198-201, doi: 10.1109/IIAI-AAI.2017.51.

- M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, and S. Ali, ‘‘Using machine learning to predict student difficulties from learning session data,’’ Artif. Intell. Rev., vol. 52, no. 1, pp. 381–407, Jun. 2019.

- O. E. Aissaoui, Y. E. A. El Madani, L. Oughdir, and Y. E. Allioui, ‘‘Combining supervised and unsupervised machine learning algorithms to predict the learners’ learning styles,’’ Procedia Comput. Sci., vol. 148, pp. 87–96, Jan. 2019.

- K. S. Rawat and I. Malhan, ‘‘A hybrid classification method based on machine learning classifiers to predict performance in educational data mining,’’ in Proc. 2nd Int. Conf. Commun., Comput. Netw. Chandigarh, India: National Institute of Technical Teachers Training and Research, Department of Computer Science and Engineering, 2019, pp. 677–684.

- N. Z. Zacharis, ‘‘A multivariate approach to predicting student outcomes in Web-enabled blended learning courses,’’ Internet Higher Edu., vol. 27, pp. 44–53, Oct. 2015.

- J. Kuzilek, M. Hlosta, and Z. Zdrahal, ‘‘Open university learning analytics dataset,’’ Sci. Data, vol. 4, no. 1, Dec. 2017, Art. no. 170171.
- A. Behr, M. Giese, and K. Theune, ‘‘Early prediction of university dropouts—A random forest approach,’’ J. Nat. Stat., vol. 1, pp. 743–789, Feb. 2020.
- https://www.kaggle.com/code/vchauhan12347/exploratory-data-analysis-oulad-data
- https://chat.openai.com