Improving Students' Performance by Interpretable Explanations using Ensemble Tree-Based Approaches

Alexandra Vultureanu-Albiși and Costin Bădică

May 3, 2021

Improving Students' Performance by Interpretable Explanations using Ensemble Tree-Based Approaches

Alexandra Vultureanu-Albişi
Faculty of Automatics, Computers and Electronics
University of Craiova, Craiova, România
alexandra.vultureanu@edu.ucv.ro

Costin Bădică
Faculty of Automatics, Computers and Electronics
University of Craiova, Craiova, România
cbadica@software.ucv.ro

***Abstract*—The careful analysis and evaluation of students' results are an important part of the educational activity, with a potentially strong impact on the students' future development. Seven classification algorithms, which are *Decision Tree*, *Bagging*, *Random Forest*, *AdaBoost*, *Gradient Boosting*, *XGBoost*, and *LightGBM*, were used in this research. In this paper, for our experiments we used two datasets, the first refers to classify and predict Portuguese language performance and the second for students' level at courses. In this paper, we propose to identify the most appropriate classification technique to improve the prediction of students' performance, interpreting it using the LIME algorithm. The obtained results using both datasets show that the model built using Decision Tree, outperforms the other constructed models. Our methodology consists of four major steps: i) analyzing and preprocessing the dataset; ii) optimizing the models using cross-validation and hyperparameter tuning; iii) comparing the performance of different ensemble tree-based models, and iv) interpreting the model by providing explanations. The development of explainable models can lead to important advantages: the model can be trusted, the transparency of the model helps to understand the underlying mechanisms that make the model work and opaque models can be interpreted without sacrificing their predictive performance.**

***Index Terms*—ensemble learning, interpretable explanation, educational data mining**

I. Introduction

Data mining methods have applications in many fields including: business (for example, marketing, stock markets), engineering (for example, fault diagnosis of rotating machines and electronic devices), and medicine (for example, medical image understanding). *Educational data mining (EDM)* is a set of techniques that can be applied to the analysis of educational data.

Academic examination plays a significant role in the academic life of any student. Predicting whether or not a student will pass the exam is the most important factor in the educational process. In addition to the fact that students will pass or fail, it also matters the level they have based on their marks, low-level, middle-level, or high-level.

Predicting a student's academic outcome requires the careful consideration of many parameters. Data regarding the student's basic knowledge about the subject, the parent accountable for the student, family support, number of school absences, and extracurricular activities can play an important role in predicting the student performance.

Students' achievements are important in the personal development of young people both to successfully integrate into society and to have a flourishing career. Besides, academic success is important because people will need higher education to be able to interact in the future with technologically advanced intelligent systems.

Predicting students' academic performance represents a real academic challenge, especially because of the additional difficulties caused by coronavirus pandemic [1]. The COVID-19 pandemic has affected education in several ways, as government has pursued actions aiming for the standard goal of reducing the spread of the coronavirus through measures that limit social contact.

Various studies of classification models were made in contemporary real-life cases. Besides their use in education, we can exemplify other areas in which prediction is necessary [14], [15], [16], [17]

Classification methods can be applied to educational data to predict students' performance. This prediction will help to proactively identify weaker students to help them to get better grades such that to have a better future. The most known prediction methods are classification, regression, and density estimation. This paper focuses on classification techniques, and the predicted variable for our datasets is a binary or categorical variable. Classification techniques can be used to predict a student's behavior in an educational environment, his interest in a subject, as well as his exam result. Furthermore, this paper aims toward improving the outcome of ensemble tree-based algorithms by adding a grid search function to boost classification accuracy in predicting the students' performance.

In modern data science, it is nowadays really important to be able to trust and understand the prediction models in addition to the increased accuracy of their produced results. Lack of transparency can sometimes result in potentially weak models with misleading conclusions.

LIME (Local Interpretable Model-Agnostic Explanations) is an algorithm designed to explain a prediction by local approximation with an interpretable model of any classifier or regressor. It acts as an *explainer* to explain the predictions in each data sample. A prediction is explained by presenting textual or visual results and supports the qualitative understanding of the relationship between the components of an instance and the prediction of the model. The LIME result is a set of explanations that represent the contribution of each feature to a prediction for one sample, which is a form of local interpretability.

LIME is a post-hoc interpretability approach, which means that the generated explanations for the predictions that were made by a trained black- box model are made after training. The main limitation of this method is that it is strictly limited to local explanations of model prediction, while it might be often useful to know the global contribution of features rather than only of the individual instances.

In this paper, we present our results of comparing models based on the ensemble tree to improve the prediction of student performance by interpreting passing or failing an exam and the level they have based on their grades. This paper aims to identify the most suitable model classification technique to improve the prediction of student performance. This study is built on several research questions:

- What are the suitable ensemble classifiers and classification techniques to improve students' performance prediction models?
- Do tuning hyperparameters help in improving ensemble classifier performance after applying the grid search technique?
- To what extent do interpretable explanations help predict students' performance?

The paper is structured as follows. In Section II we briefly present a review of related works. Then the paper follows with Section III devoted to describing the methodology adopted for this research. The experiments, results, and brief exposure of the software packages used in our research are presented in Section IV. Finally, in Section V we present our conclusions and future works.

II. Related Works

Due to the massive amounts of information generated within the field of education, there has been a rise in interest in developing educational data mining techniques. This interest led to several international research EDM conferences held after 2007, as well as to the establishment of an academic journal in 2009, the *Journal of Educational Data Mining*. A first review of the state-of-the-art of EDM is presented in the paper [2].

Authors of the paper [7] applied two decision tree methods, *RandomTree* and *REPTree*, for EDM. The performance of a decision tree was optimized by using an ensemble technique named *Rotation Forest* algorithm. Their experimental outcomes showed that the ensemble decision tree methods can create more understandable rules than simple decision trees and they can be used for educational data mining due to their good performance.

An interesting approach was proposed in [8] to analyze and identify the impact of student background, student social activities, and student coursework achievement in predicting student academic performance. In this research, the authors used supervised educational data mining techniques, *Naïve Bayesian*, *Multilayer Perceptron*, *Decision Tree J48*, and *Random Forest* for predicting students' mathematics performance in secondary school.

Authors of the study [9] applied both supervised and unsupervised machine learning techniques to discover which significant features usually characterize successful learning in a computing course. The used algorithms are *Decision Tree*, *Bayesian Network*, *Naïve Bayes*, *Support Vector Machine*, *Multilayer Perceptron*, *Logistic Regression*, *K-Nearest Neighbor*, and *Association Rule*.

In paper [10] authors proposed principal component analysis and relational association rule mining, as intelligent tools for academic data sets analysis. They used in their experiments a real data set, which contained the grades received by students who took a Computer Science undergraduate course offered by Babeş-Bolyai University from România. Ensemble techniques are used to empower computation, functionality, robustness, and accuracy aspects of machine learning models. Authors of the paper [13] presented the state-of-the-art of these techniques.

The research of paper [11], was focused on the role of AI/ML in education, for predicting student performance. It also focused on the need for human interpretable model results, providing an insight into the factors that influence prediction.

Another approach of using interpretable explanations for students' performance was proposed in paper [12]. Authors compared the performance of *LightGBM* with *XGBoost*, *Random Forest*, and *Decision Trees* and using *SHapley Additive exPlanations (SHAP)*, to interpret and visualize the contribution of features for educational data from South Africa.

III. Methodology

A. Problem Description

In this paper, we propose to optimize the classification models by adjusting the hyperparameters that define the model shape and structure. This means that we are interested in determining the values of the hyperparameters to obtain the best predictions using the data sets from our experiments. Here, we experimentally evaluate the performance of seven models based on the ensemble tree to predict whether or not a student passes an exam or whether if a student is low-level, middle-level, or high-level.These include *Decision Tree*, *Bagging*, *Random Forest*, *AdaBoost*, *Gradient Boosting*, *XGBoost*, and *LightGBM* algorithms, as shown in our proposed framework presented in Figure 1.

Cross-validation was used to accurately estimate how well our trained models perform on testing data. All the models were trained and tested on the same training dataset.

Finally, we used LIME to interpret our model such that it can be easily perceived and affirmed by users. LIME method interprets an individual prediction by learning a local interpretable model [3]. The idea behind LIME is that it samples instances both in proximity and far from the interpretable representation of the original entry. LIME uses the interpretable representation of these samples, shows their predictions, and constructs a weighted linear model by minimizing losses and complexity. The weights of the points decrease as the points are farther away. The explanation at the local level is accurate, which means that it represents the model prediction of the neighboring instances.

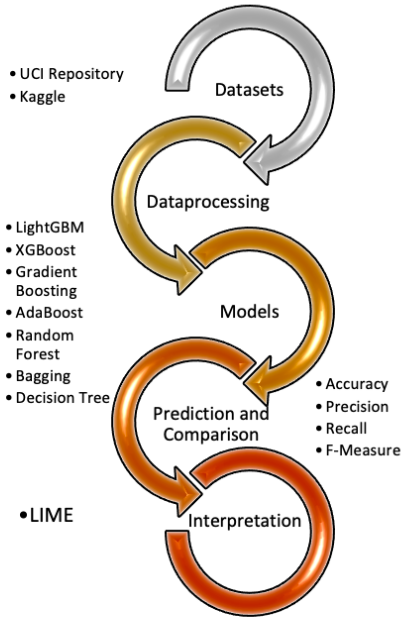

Fig. 1. The proposed framework

B. Datasets Description

A challenge faced by EDM research may be caused by the limited access to educational data sources. So in our research, we used different existing datasets from publicly available archives. Preprocessing is an essential step in data preparation, to increase data quality and to ensure that the modeling process is more efficient. Data preprocessing is always a necessary preliminary step in EDM, before determining the models for predicting students' performance in higher education.

Our first proposed dataset was obtained from the *UCI Machine Learning Repository* [4]. This dataset includes 33 features and 649 instances. The features are classified into student grades, demographic, social, and school related features. This data set represents the results of students in secondary education in two Portuguese schools and it provides the performance of students in a Portuguese language subject. It includes both binary and numerical values.

Features of the data include students' grades, as well as students' demographic, social, and school related information. The target feature *G3*, the final year grade, has a strong correlation with features *G1* and *G2*, which correspond to the 1st and 2nd grading periods. It is more difficult to make the prediction using only *G3*, without *G2* and *G1*. Therefore, we propose to represent the target class using the *performance* feature, created by adding the 3 features *G1*, *G2*, and *G3* such that if the resulting sum is greater than 35 this means that the student has passed, and if it is less than 35 then he or she has failed. This brought us to an even division of about 50% of students who passed and failed, giving us a balanced dataset using only the remaining 31 features. After setting the *performance* feature as the target class in the first dataset, we obtained 328 entries labeled as *Pass* and 321 entries labeled as *Fail*.

Our second proposed dataset was obtained from *Kaggle*. It is an educational dataset that was collected from a learning management system [5], [6]. This dataset includes 17 features and 480 instances. It includes both nominal and numerical values. The dataset was collected in two educational semesters and course topics were: English, Spanish, French, Arabic, IT, Maths, Chemistry, Biology, Science, History, Quran, Geology. The features are classified into demographic features, such as gender and nationality, academic background features such as educational stage, grade level, and behavioral features such as raising a hand in class opening resources, parent satisfaction with the progress of their children during their educational activity. The students are classified into three numerical intervals based on their total grade: *Low-Level* (includes values from 0 to 69), *Middle-Level* (includes values from 70 to 89), and *High-Level* (includes values from 90-100).

Both data sets were cleaned and we noticed that they had no missing (*NA*) values. The first step we took was to convert features into categorical types. In Python, a good practice is to capture categorical features using the Pandas's "category" *dtype*, as this makes the processing of Pandas's *++DataFrame* column operations much faster than with "object" *dtype*. Since we need the numerical representation of any categorical feature, we use the *label encoding scheme* to encode each value of a column to a number. Numerical labels are always between 0 and the $number\_of\_features - 1$.

Finally, we did the data sets splitting, keeping 70% for training and 30% for testing. Then, the 5-fold cross-validation technique was applied to each classification model. Grid search technique was used to find the best hyperparameters for each model, to get the best classification result.

C. Classification Techniques

In this paper, we used six different techniques of ensemble learning. By default, all ensemble learning classifiers are using the *Decision Tree* method as a base algorithm. Furthermore, we added the standard *Decision Tree* method to the initial list. A brief description of the utilized classifiers is as follows:

1) *Decision Tree* – a hierarchical representation of possible solutions to a decision problem based on checking a sequence of conditions. The decision tree algorithm is frequently used for classification tasks. Decision trees classify data from data sets into specified classes based on the values of input variables. We used *scikit-learn*

which provides an optimized version of the *CART (Classification and Regression Trees)* algorithm. CART is very similar to *C4.5*, but differs in that it accepts numeric target variables and does not calculate sets of rules, but builds binary trees using the feature and threshold that produce the highest information gain at each node.

2) *Bootstrap aggregating (Bagging)* – is a simple ensemble learner that combines basic models for construction and aggregation. The basic models are created using bootstrap samples of the training set and voting or average calculation for prediction.
3) *Random Forest* – is a bagging algorithm that uses ensemble learning techniques by training every tree independently and collecting different decision trees whose results are aggregated into a single final result by voting mechanisms.
4) *AdaBoost* – is one of the most popular algorithms for classification because of its excellent performance. The main idea is to construct a succession of weak learners through different training sets with different weights.
5) *Gradient Boosting* – it employs a gradient descent algorithm to minimize errors in sequential models.
6) *The extreme Gradient Boosting (XGBoost)* - is a powerful ensemble learning technique and an optimized gradient boosting algorithm through parallel processing, handling missing values, and using regularization to avoid overfitting.
7) *LightBGM* – is a gradient boosting framework that uses tree-based learning algorithms and outperforms *XGBoost*. It has the following advantages: faster training speed and higher efficiency, better accuracy, and lower memory usage. It will sacrifice a certain accuracy of the model and increase the training time, but it can improve the interpretability of the model.

IV. Results and Discussions

A. Libraries and Tools

This study used *Python 3.7.4* language and *Jupyter Notebook* to execute the experiment. The *Scikit-learn*, *Pandas*, and *Numpy* packages were used to preprocess the datasets and to apply ensemble tree-based methods in the proposed model. For *XGBoost* and *LightGBM* classifiers we used separate Python packages, *XGBoost* and *LightBGM*. For local interpretable model-agnostic explanations we used the *Lime* package.

The Python package *Pandas* was used for importing data from *CSV* files into a specialized data structure called a *data frame*. The data set split ratio was achieved by using the *train_test_split* method of the *model_selection* module of *Scikit-learn*. For hyperparameter tuning, we used the *GridSearchCV* member function of the *model_selection* module from *Scikit-learn* package.

B. Testing the Performance and Discussion of Results

In our experiments, we have used 5-fold cross-validation. While designing the model, we evaluated the impact of multiple values of the models' hyperparameters that control the training process. They are presented in Table I.

In this research, the algorithms were applied to two data sets. To evaluate our models, we observed the improvement of the accuracy by comparing the accuracy obtained using default values for hyperparameters with the accuracy provided by models after hyperparameter tuning using the grid search technique. Our obtained results are shown in Table II. As we can see, on each dataset, the algorithms had improvements after the application of hyperparameter tuning using grid search. However, we would like to point out that some models have not undergone an increased accuracy. Such as *Gradient Boosting* for both datasets, *XGBoost* for the first dataset and, *AdaBoost* for the second one.

The best accuracy was given by *Gradient Boosting* and *XGBoost* for the first dataset with 73%. For the second dataset, the best accuracy was given by *Bagging* and *XGBoost* with 69%. The improvement in accuracy is not very great, but even so, we can get a better prediction for the models.

C. LIME Use Case Scenario

To improve the transparency of models along with LIME, we chose to exemplify the prediction for an instance of each test set from both datasets, and thus we will check if the explanation improves the transparency of classification and determines with what probability the instance is correctly classified. We mention that the instance chosen from the first dataset is mapped to the *Pass* class, while the instance chosen from the second dataset was mapped to the *High-Level* class. The results are presented in Table III, where we can conclude that the chosen instance was classified with a high probability, both for the first dataset and for the second using the *Decision Tree*, *XGBoost*, and *Gradient Boosting* models. From the results in the previous table, we can also see that the *AdaBoost* model is not efficient for class prediction for any of the datasets.

LIME helps us to answer the question: *what exactly caused the prediction to be like this?*. LIME modifies the data sample by slightly modifying the values of the features and collects the resulting impact of each feature change to the prediction.

LIME generates explanations for a prediction by returning an *explanation object* using *explain_instance method*, which converts the local linear model's predictions from numerical form to a visual, interpretable form. The interpretable form returns the prediction given by the model for the given test vector (instance) and the local prediction that returns the values obtained by the model trained on the modified features and using only the top features outputted by LIME.

The prediction of the first dataset was a binary classification problem and we were able to generate and visualize the features offered by LIME that helped the prediction. To illustrate what was the contribution of features to the class prediction we presented in Figure 4 the features that were generated with LIME using *Decision Tree*.

To predict the class for the first dataset and using explanation object, we showed only the first 10 features that helped

TABLE I
CONSIDERED VALUES OF HYPERPARAMETERS

Models	**Hyperparameters**	**Default values**	**Considered values**	**Best parameters First Dataset**	**Best parameters Second Dataset**
DecisionTree	max_depth	$None$	$3, 5, 7, 9$	9	5
Bagging	$n\_estimators$	10	$10, 50, 70, 90$	90	90
RandomForest	$n\_estimators$	100	$100, 300, 500$	300	300
	max_depth	$None$	$3, 5, 7, 9$	15	9
AdaBoost	$learning\_rate$	1	$0.01, 0.1, 1$	0.1	1
GradientBoosting	$learning\_rate$	0.1	$0.01, 0.1, 1$	0.1	0.1
XGBoost	max_depth	6	$3, 4, 5, 6, 7, 8, 9, 10$	3	10
LightGBM	max_depth	-1	$25, 50, 75$	25	25
	$learning\_rate$	0.1	$0.01, 0.05, 0.1$	0.01	0.05
	$min\_data\_in\_leaf$	20	$25, 30, 35$	30	30
	num_leaves	31	$40, 50, 60, 70$	40	40

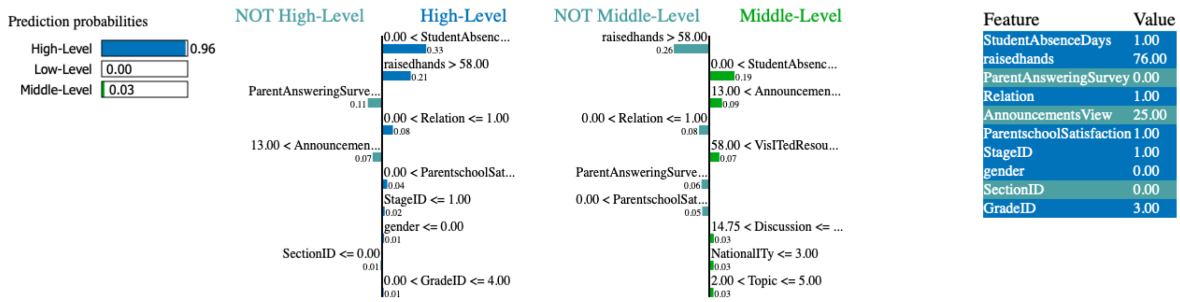

Fig. 2. Explanation object for an instance from second dataset using *XGBoost*

TABLE II
ACCURACY BEFORE AND AFTER GRIDSEARCH

	First Dataset Accuracy		**Second Dataset Accuracy**	
Models	**Default Values**	**With GridSearch**	**Default Values**	**With GridSearch**
DecisionTree	0.63	0.67	0.60	0.63
Bagging	0.64	0.71	0.66	0.69
RandomForest	0.67	0.71	0.63	0.68
AdaBoost	0.69	0.72	0.54	0.54
GradientBoosting	0.73	0.73	0.66	0.66
XGBoost	0.73	0.73	0.63	0.69
LightGBM	0.68	0.70	0.65	0.67

predicted the target class in Figure 3 where the *Decision Tree* classifier predicted that the chosen instance belongs to the *Pass* class and it was very confident about this prediction because the probability for this class was 0.90.

The prediction of the second dataset was a multi-class classification problem and to illustrate an explanation object for this dataset we presented in Figure 2 the *XGBoost* model. The *XGBoost* classifier predicted that chosen instance from the second dataset belongs to the *High-Level* class and it was very confident about this prediction because the probability for this class was 0.96. And LIME explained that the classifier assigned this class based on the features shown in Figure 2 where we presented only the first 10 features that helped predicted the target class.

TABLE III
PREDICTION PROBABILITY FOR TEST INSTANCE

	First dataset Prediction probabilities		Second dataset Prediction probabilities		
Models	**Pass**	**Fail**	**High**	**Middle**	**Low**
DecisionTree	0.90	0.10	0.91	0.09	0.00
Bagging	0.88	0.12	0.73	0.27	0.00
RandomForest	0.75	0.25	0.73	0.26	0.01
AdaBoost	0.53	0.47	0.37	0.37	0.25
GradientBoosting	0.85	0.15	0.86	0.14	0.00
XGBoost	0.84	0.16	0.96	0.03	0.00
LightGBM	0.84	0.16	0.68	0.32	0.00

V. CONCLUSIONS AND FUTURE WORKS

Our paper aimed to experimentally evaluate and compare the results obtained by using ensemble tree-based to classify students' performance, using hyperparameter adjustment to improve accuracy. In addition to improving accuracy, we proposed interpreting the results of the models using the LIME technique. In response to the questions asked at the beginning of this research, we present them in the following.

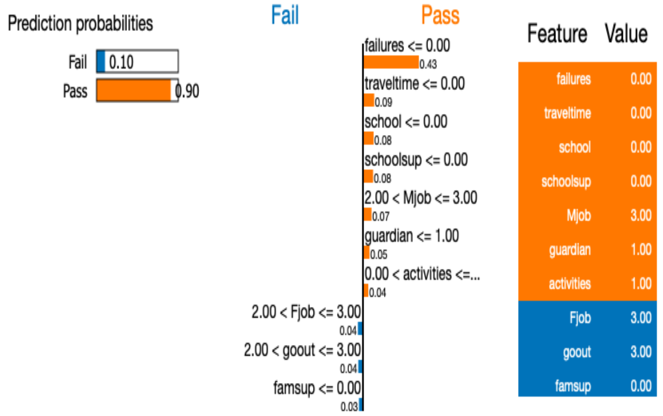

Fig. 3. Explanation object for an instance from first dataset using *Decision Tree*

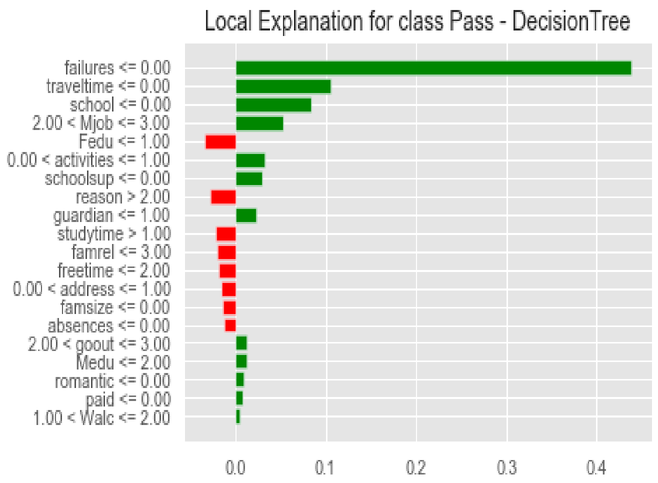

Fig. 4. Important features generated with LIME for first dataset

- In this paper, we used ensemble classifiers because are one of the well-established models to increase the confidence in the prediction. Following the experiments, we can say that the ensemble classifiers and classification techniques can improve students' performance prediction based on the datasets presented above.
- We noticed that tuning hyperparameters help in improving ensemble classifier performance after applying the GridSearch technique.
- As a result of both datasets, interpretable explanations provided help to trust in a model with a human-understandable explanation.

As future work we would like to expand this experiment using other interpretable and classification methods and also to use a dataset with students' performance from the University of Craiova.

Many studies, showing that the EDM provides an effective prediction of students' performance. Besides, for our future work, we would like to advance in this field by using recommender systems that use human-understandable explanations.

REFERENCES

[1] T. Gonzalez, M. A. de la Rubia, K. P. Hincz, M. Comas-Lopez, Laia Subirats, Santi Fort and G. M. Sacha, "Influence of COVID-19 confinement on students' performance in higher education," in PloS one 15, no. 10, 2020.

[2] C. Romero and S. Ventura, "Educational Data Mining: A Review of the State of the Art," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 6, pp. 601-618, Nov. 2010, doi: 10.1109/TSMCC.2010.2053532.

[3] M. T. Ribeiro, S. Singh and C. Guestrin, "" Why should I trust you?" Explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016.

[4] P. Cortez and A. Silva, "Using Data Mining to Predict Secondary School Student Performance," in A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, ISBN 978-9077381-39-7, 2008.

[5] E. A. Amrieh, T. Hamtini and I. Aljarah, "Mining Educational Data to Predict Student's academic Performance using Ensemble Methods," in International Journal of Database Theory and Application, 9(8), pp. 119-136, 2016.

[6] E. A. Amrieh, T. Hamtini and I. Aljarah, "Preprocessing and analyzing educational data set using X-API for improving student's performance," in Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on pp. 1-5, 2015.

[7] M. Abdar, M. Zomorodi-Moghadam and X. Zhou, "An Ensemble-Based Decision Tree Approach for Educational Data Mining," 2018 5th International Conference on Behavioral, Economic, and Socio-Cultural Computing (BESC), Kaohsiung, Taiwan, 2018, pp. 126-129, doi: 10.1109/BESC.2018.8697318.

[8] C. Kiu, "Data Mining Analysis on Student's Academic Performance through Exploration of Student's Background and Social Activities," 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA), Subang Jaya, Malaysia, 2018, pp. 1-5, doi: 10.1109/ICACCAF.2018.8776809.

[9] C. -Y. Ko and F. -Y. Leu, "Examining Successful Attributes for Undergraduate Students by Applying Machine Learning Techniques," in IEEE Transactions on Education, vol. 64, no. 1, pp. 50-57, Feb. 2021, doi: 10.1109/TE.2020.3004596.

[10] L. M. Crivei, G. Czibula, G. Ciubotariu and M. Dindelegan, "Unsupervised learning based mining of academic data sets for students' performance analysis," 2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2020, pp. 000011-000016, doi: 10.1109/SACI49304.2020.9118835.

[11] M. Chitti, P. Chitti, and M. Jayabalan, "Need for Interpretable Student Performance Prediction," in Conference: 2020 13th International Conference on Developments in eSystems Engineering (DeSE), december 2020.

[12] H. Wandera, M. Vukosi, and D. Sengeh, "Investigating similarities and differences between South African and Sierra Leonean school outcomes using Machine Learning." arXiv preprint arXiv:2004.11369 (2020).

[13] Ardabili, Sina, Amir Mosavi, and Annamária R. Várkonyi-Kóczy. "Advances in machine learning modeling reviewing hybrid and ensemble methods." In International Conference on Global Research and Education, pp. 215-227. Springer, Cham, 2019.

[14] Choubin, B., Borji, M., Hosseini, F.S., Mosavi, A. and Dineva, A.A., 2020. Mass wasting susceptibility assessment of snow avalanches using machine learning models. Scientific reports, 10(1), pp.1-13.

[15] Mosavi, A., Sajedi-Hosseini, F., Choubin, B., Taromideh, F., Rahi, G. and Dineva, A.A., 2020. Susceptibility mapping of soil water erosion using machine learning models. Water, 12(7), p.1995.

[16] Mosavi, A., Golshan, M., Janizadeh, S., Choubin, B., Melesse, A.M. and Dineva, A.A., 2020. Ensemble models of GLM, FDA, MARS, and RF for flood and erosion susceptibility mapping: a priority assessment of sub-basins. Geocarto International, pp.1-20.

[17] Mosavi, A., Hosseini, F.S., Choubin, B., Goodarzi, M., Dineva, A.A. and Sardooi, E.R., 2021. Ensemble Boosting and Bagging Based Machine Learning Models for Groundwater Potential Prediction. Water Resources Management, 35(1), pp.23-37.