Using Data Mining for Predicting Student Academic Performance

Manjusha Awasthi, Logan McMillan, Yu Zhuang
Department of Computer Science
North Carolina State University
Raleigh, NC-27606

Sakib Zargar
Department of Mechanical and Aerospace Engineering
North Carolina State University
Raleigh, NC-27606

Travis Martin
Department of Education
North Carolina State University
Raleigh, NC-27606

***Abstract*—Educational Data Mining (EDM) is an emerging interdisciplinary research field which involves exploring large scale datasets that come from educational settings in order to better understand students and the settings in which they learn. This is done with the aim of being able to predict a student's performance and provide suggestions for improving it. As such, in this project, the objective is twofold: First, to predict student performance given their personal and socio-economic attributes. Second, to identify the key factors that affect the performance. The dataset used is the publicly available student performance data set from the UCI Machine Learning Repository.**

I. INTRODUCTION

Modelling student performance is an important tool for both educators and students as it can help in better understanding the overall education system and ultimately improving it. For instance, school professionals could perform corrective measures for weak students (e.g. remedial classes, more interventions etc.,). However, predicting academic performance of students is challenging as it depends on diverse factors such as personal, socio-economic, psychological and other environmental variables, many of which cannot be easily categorized. In this work, we will analyze real-world data from two Portuguese secondary schools [1]. The dataset used is the publicly available student performance data set from the UCI Machine Learning Repository. The data attributes include student grades, demographic, social and school related features and was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por).

II. RELATED WORK

Educational research is still very much an emerging field of study in data science. This may be due to several factors, but most likely it is due to less money being available to pursue these endeavors. Mukesh Kumar and A. J. Singh [2] applied different data mining algorithms on a student dataset to predict student performance. Satyanarayana, Ashwin & Nuckowski, M. [3] used ensemble filtering on the Portuguese secondary schools' dataset to try to improve upon the results obtained in [1]. T. Devasia, Vinushree T P and V. Hegde [4] created an online system using Naïve Bayesian techniques in order to predict student performance.

Currently, a lot of research and effort is being put into something called value added achievement [5]. Value added achievement takes students with similar testing history, groups them and then uses this information to predict the future success of these students based on a single variable, the student's teacher or school. If a school or teacher with these students is able to make them do better than the prediction, the school or teacher is said to have added value to their learning.

III. OUR CONTRIBUTION

Starting with [1] as the baseline, the first step was to replicate the results from the paper. This was done both for the 2-level and 5-level classification scheme. In the original implementation, the importance of the features was established by training multiple models by leaving out features one at a time (Scheme: A, B and C) and seeing the impact on the overall model performance. This is an inefficient way of determining feature quality especially when the number of input features is large. In this study, an efficient implementation was used for feature selection. Also, data mining algorithms like Logistic Regression,

Naïve Bayes and certain Ensemble techniques which weren't implemented in the original paper were tested to check their performance on the given dataset.

After implementing all the above mentioned models using Python inbuilt functions, we also implemented two ML models (ANN and DT) from scratch and tested their performance on the given dataset.

IV. Implementation And Results

A. *Replicating results from the baseline paper:*

The aim is to model the final test score (G3) using the attributes given in the dataset. G3 is a discrete variable with values ranging from 0-20. In the original paper the 2-level classification involves diving the scores into 0 (Scores: 0-9) and 1 (Scores: 10-20) while as the 5-level classification involves diving the scores into 0 (Scores: 0-9), 1 (Scores: 10-11), 2 (Scores: 12-13), 3 (Scores:14-15), 4 (Scores:16-20). The distributions for the two datasets for the 2-level and 5-level classification tasks are given in Figures 1 and 2 respectively.

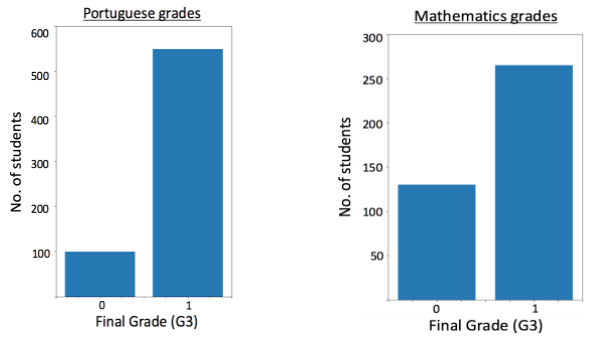

Fig 1: Grade distribution for 2-leavel classification case

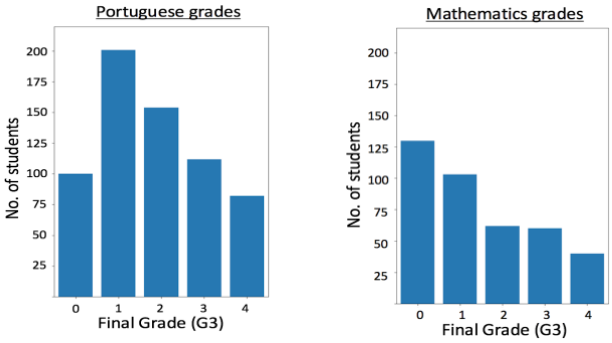

Fig 2: Grade distribution for 5-leavel classification case

The adjoining tables give the comparison of the results for various ML models with the baseline implementation. The nomenclature used in the tables is as follows:

- LR: Logistic Regression. Implemented using LogisticRegression from SciKit learn.
- NB: Naïve Bayes classification. Implemented using GaussianNB from SciKit learn.
- NN: Artificial Neural Network. Implemented using MLPClassifier from SciKit learn.
- RF: Random Forest classifier. Implemented using RandomForestClassifier from SciKit learn.
- MV: Ensemble Majority Vote classifier. Implemented using VotingClassifier with SVC, DT and RF classifiers as the base classifiers in SciKit learn.
- BG: Bagging classifier. Implemented using BaggingClassifier with DT as the base classifier in SciKit learn.
- DT: Decision Tree classifier. Implemented using DecisionTreeClassifier from SciKit learn.
- SVM: Support Vector Machine classifier. Implemented using SVC from SciKit learn.

TABLE 1: 2-LEVEL CLASSIFICATION RESULT COMPARISON

	Portuguese dataset		**Mathematics dataset**	
	Baseline results	Our results	Baseline results	Our results
LR	-	**91.2**	-	**90.4**
NB	-	**85.7**	-	**79.2**
NN	**90.7**	89.4	88.3	**90.1**
RF	**92.6**	91.8	91.2	91.2
MV	-	**92.4**	-	**90.4**
BG	-	**91.1**	-	**88.4**
DT	**93.0**	92.4	90.7	**91.9**
SVM	**91.4**	88.6	86.3	**89.1**

TABLE 2: 5-LEVEL CLASSIFICATION RESULT COMPARISON

	Portuguese dataset		**Mathematics dataset**	
	Baseline results	Our results	Baseline results	Our results
LR	-	**53.3**	-	**58.7**
NB	-	**46.9**	-	**54.4**
NN	**65.1**	63.0	60.3	**64.1**
RF	**73.5**	70.9	**72.4**	70.1
MV	-	**70.6**	-	**70.9**
BG	-	**72.0**	-	**73.4**
DT	76.1	76.0	**76.7**	70.6
SVM	64.5	**69.6**	59.6	**69.6**

The values missing in the tables for the baseline results correspond to the models that were not implemented in the original paper. As can be seen from the tables, the results obtained were comparable to the baseline implementation.

Next, the features were ranked according to their importance using their absolute covariance values with the final score G3. Figures 3 and 4 give the results for the two datasets. In both the cases the top 5 features were selected and the ML models were run for the selected features only (just the 5-level classification case). Table 3 gives the comparison of the results before and after feature selection.

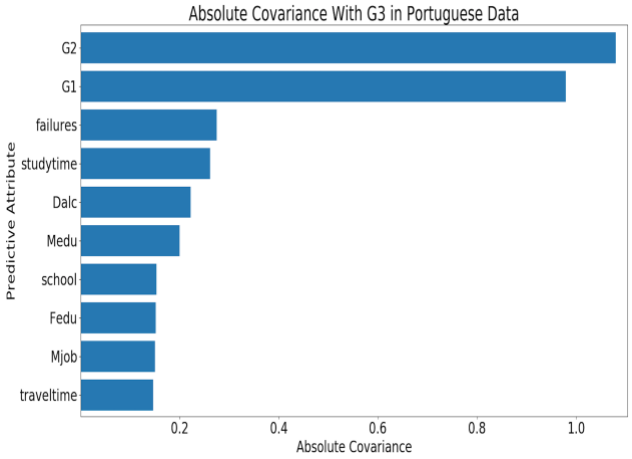

Fig 3: Feature ranking for the Portuguese dataset

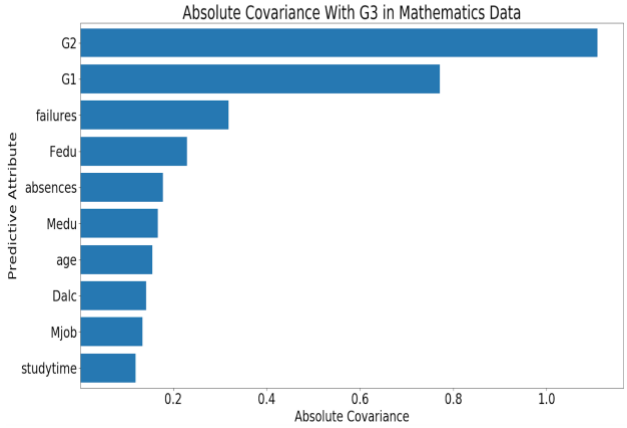

Fig 4: Feature ranking for the Mathematics dataset

TABLE 3: COMPARING RESULTS AFTER AND BEFORE FEATURE SELECTION

	Portuguese dataset		**Mathematics dataset**	
	5-features only	All features	5-features only	All features
LR	**63.2**	53.3	**67.1**	58.7
NB	**65.0**	46.9	**71.6**	54.4
NN	**64.1**	63.0	**71.4**	64.1
RF	66.4	**70.9**	**71.1**	70.1
MV	67.0	**70.6**	**71.9**	70.9
BG	66.4	**72.0**	68.8	**73.4**
DT	76.0	76.0	**72.7**	70.6
SVM	**71.8**	69.6	**72.7**	69.6

Interestingly, the individual classifiers (LR, NB, NN, DT, SVM) all perform better after feature selection whereas the performance of the ensemble methods seems to deteriorate.

In an attempt to further improve the accuracy, the scores G1 and G2 were also split into 5 levels like the G3 score. The results are presented in Table 4.

TABLE 4: COMPARING RESULTS AFTER AND BEFORE DISCRETIZING G1 AND G2

	Portuguese dataset		**Mathematics dataset**	
	Without discretizing G1 and G2	After discretizing G1 and G2	Without discretizing G1 and G2	After discretizing G1 and G2
LR	63.2	**64.3**	**67.1**	66.3
NB	**65.0**	64.7	71.6	**72.9**
NN	64.1	**65.9**	**71.4**	69.4
RF	66.4	**69.5**	**71.1**	69.6
MV	67.0	**70.3**	**71.9**	68.4
BG	66.4	**67.2**	68.8	**70.1**
DT	**76.0**	75.7	72.7	**78.5**
SVM	71.8	**75.7**	72.7	**78.2**

B. *Implementing ANN from scratch*

In recent times, Artificial Neural Networks have gained importance especially in the field of deep learning where-in deep neural networks have been seen to constantly outperform traditional ML algorithms. Figure 5 shows a NN with 2 hidden layers and a single neuron in the output layer. Such a set-up could be used for a binary classification problem.

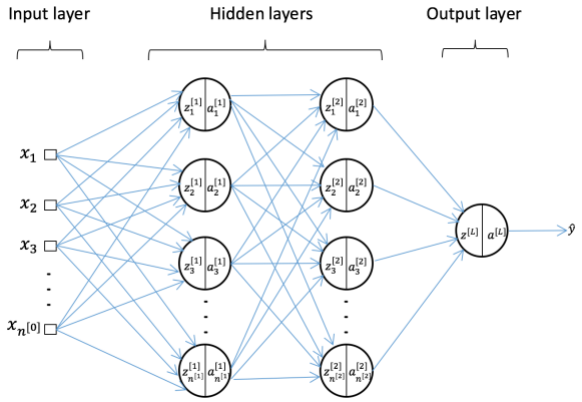

Fig 5: 3-layer ANN (2-hidden layers and 1 output layer)

The NN nomenclature used is as follows:

- L : Total number of layers in the ANN.
- $n^{[0]}$: Number of input features.
- m : Number of training examples.
- $n^{[l]}$: Number of hidden units in layer *l*.
- $W^{[l]}$: ($n^{[l]} \times n^{[l-1]}$) dimensional weight matrix associated with layer *l*.
- $b^{[l]}$: ($n^{[l]} \times 1$) dimensional bias vector associated with layer *l*.
- $Z^{[l]}$: $n^{[l]} \times m$ matrix of pre-activation values for layer *l*.
- $g^{[l]}(.)$: Activation function for layer *l*.
- $A^{[l]}$: $n^{[l]} \times m$ matrix of post-activation values for layer *l*.

Eqs. (1) and (2) represent the forward pass (calculation of pre-activation and post-activation values for each layer) while as Eqs. (3), (4), (5) and (6) represent the backpropagation pass (updating the weights and biases using gradient descent algorithm).

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]} \tag{1}$$

$$A^{[l]} = g^{[l]}(Z^{[l]}) \tag{2}$$

$$dZ^{[l]} = dA^{[l-1]} * g^{[l]'}(Z^{[l]}) \tag{3}$$

$$dW^{[l]} = \frac{1}{m} dZ^{[l]} dA^{[l-1]^T} \tag{4}$$

$$db^{[l]} = \frac{1}{m} \sum^{l} dZ^{[l]} \tag{5}$$

$$dA^{[l-1]} = W^{[l]^T} dZ^{[l]} \tag{6}$$

Note: The implementation is done for the 2-level classification case for the two datasets. As such, the network has only one node in the output layer. The hyper-parameters that the user can control in the implementation are: network architecture ($[n^{[0]}, n^{[1]}, n^{[2]}, \ldots, n^{[L]}]$), activation function for the hidden layers (sigmoid, tanh, relu), learning rate and the number of iterations. Figure 6 represents the decay in the cost function with the number of iterations for the Portuguese dataset.

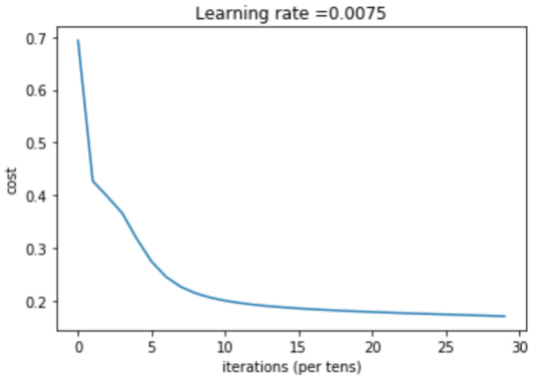

Fig 6: Decay in cost function with the number of iterations.

The results obtained look better than the SciKit Learn implementation but that's mainly because here a simple train-test split is done rather than performing a k-fold cross validation.

TABLE 5: 2-LEVEL CLASSIFICATION RESULTS FROM NN IMPLEMENTATION

	Training accuracy	Test accuracy
Portuguese language	94 %	90 %
Mathematics	95 %	93 %

C. Implementing Decision Tree from scratch

The decision tree implementation is an object-oriented implementation based on the divide-and-conquer C4.5 tree building algorithm. All categorical variables should be encoded as numerical variables prior to building the tree. It classifies this numerical data using a binary split at the optimum threshold and does not perform post-pruning. It offers two options for pre-pruning, however. The arguments max_depth and min_labels in the tree's constructor offer a way to limit the depth of the final tree and limit the number of labels required to split a node accordingly. It uses information gain ratio as its splitting criteria and uses the NumPy library for performance optimization.

The decision tree implementation is comprised of three classes. First is the DecisionTree class. DecisionTree is built similarly to the classifiers in the SciKit Learn library. It offers three methods found in the SciKit Learn classifiers. Fit is the method that constructs the tree based on the given attributes and labels. Predict uses the built model in order to predict and return the labels for the given data set. DecisionTree holds a single root node which is a DecisionDTNode. DecisionDTNode is the most important class for building the decision tree. It holds many helper methods for calculating entropy, gain ratio, and finding the appropriate threshold at any given node. The build_tree method is the most important method for building the decision tree, however. It holds the logic to build the tree recursively using the helper methods and then return the fully built tree's root. It also holds the logic for predicting class labels recursively. Finally, the LeafDTNode class is simple. It acts as a placeholder in the tree. It holds only the leaf's corresponding label and the predict method that acts as the base case for DecisionDTNode's predict function. Once the LeafDTNode's predict function is called, it returns its corresponding label.

The DT implementation is done for the 5-level classification problem, and Table 6 gives the results with 10-fold cross validation.

TABLE 6: 5-LEVEL CLASSIFICATION RESULTS FROM DT IMPLEMENTATION

	Accuracy
Portuguese language	67.3 %
Mathematics	67.6 %

V. CONCLUSION

Educational data mining has a lot of potential for improving the overall quality of education especially in developed countries where data can be efficiently collected and stored in educational settings. Also, better ways of making the data accessible for research purposes should be made without compromising on the privacy of students. The dataset that we used in the study ended up having a lot of irrelevant features which is clear by the fact that eliminating a lot of the input features improved the performance.

REPOSITORY

LINK TO GOOGLE DRIVE FOLDER:

https://drive.google.com/drive/u/0/folders/1BTD5LPOGgZ9aR3YeWjnfz_wShQdYwFAW

REFERENCES

[1] 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, 2008, EUROSIS, ISBN 978-9077381-39-7.

[2] Mukesh Kumar, A.J. Singh,"Evaluation of Data Mining Techniques for Predicting Student's Performance", International Journal of Modern Education and Computer Science(IJMECS), Vol.9, No.8, pp.25-31, 2017.DOI: 10.5815/ijmecs.2017.08.04I.

[3] Satyanarayana, Ashwin & Nuckowski, M. (2016). Data Mining using Ensemble Classifiers for Improved Prediction of Student Academic Performance.

[4] Blazar, D., Litke, E., & Barmore, J. (2016). What does it mean to be ranked a high or low value-added teacher? Observing differences in instructional quality across districts. *American Educational Research Journal*,*53*(2), 324-359.