Research Overview

After looking into the possibility of accurately predicting the performance of a student based on past data, we have come to the conclusion that, based on the data available on the application (assessments only, no course data), the J48 Decision Tree or Random Forest algorithms fit this problem best.

To support this affirmation, the following table has been extracted from a research paper[1] on the topic:

	Dataset	Precision	Recall	F-measure	Kappa	AUC
J48	All features	0.99	0.99	0.99	0.98	0.99
	Assessment only	0.99	0.99	0.99	0.98	0.99
	Assessment + course access	0.99	0.99	0.99	0.98	0.99
	Assessment + mobile course access	0.99	0.99	0.99	0.98	0.99
	Course access + mobile course access	0.94	0.93	0.93	0.86	0.94
	Course access only	0.93	0.91	0.91	0.83	0.93
	Mobile course access only	0.86	0.81	0.80	0.62	0.88
RF	All features	0.99	0.99	0.99	0.98	1
	Assessment only	0.99	0.99	0.99	0.98	1
	Assessment + course access	0.99	0.99	0.99	0.98	1
	Assessment + mobile course access	0.99	0.99	0.99	0.98	1
	Course access + mobile course access	0.97	0.96	0.96	0.93	1
	Course access only	0.97	0.97	0.97	0.94	0.99
	Mobile course access only	0.89	0.85	0.84	0.69	0.95

There are multiple research papers that also investigate the possibility of using Decision Trees as a means to predict the GPA of a student[2], but they require more data in doing so. This means that, additionally, there needs to be supplementary data about the assignments and courses.

According to some performance comparison[4] on small datasets, like the one that results from our application, whilst the Bayes Naive algorithm may offer more balanced results, the J48 algorithm gives the best result without overfitting on small datasets.

J48 Algorithm

One of the earliest decision tree algorithms is the C4.5 tree developed by Ross Quinlan[5]. The basic idea of this tree is to build other trees from a group of training data using the concept of information entropy[6].

The C4.5 tree works the same way as the ID3 (Iterative Dichotomiser 3) tree, as it is merely an extension of it. In 2011, authors of the Weka machine learning software described the C4.5 algorithm as "a landmark decision tree program that is probably the machine learning workhorse most widely used in practice to date". It soon became popular after ranking 1 in the *Top 10 Algorithms in Data Mining* paper published by Springer LNCS in 2008. However, the newer concept can handle both continuous and discrete attributes. In order to handle continuous ones, C4.5 creates a threshold and then splits the list in two categories: attribute value above the threshold and less than or equal to it. Moreover, it can also handle training data with missing attribute values, allowing them to be marked by ?, which highlights that the value is missing. These attributes are not used in gain and entropy calculations. Another improvement brought by C4.5 is that it can handle attributes with differing costs and allows trees to be pruned after the creation. This algorithm goes back through the tree after its creation and attempts to remove unnecessary branches and replace them with leaf nodes.

J48 is an open source Java implementation of the C4.5 algorithm in the WEKA. This algorithm was chosen thanks to its capabilities to handle educational datasets well and provide high accuracy results (this was highlighted in [4], [6], [7], [8]).

Since the appearance of the aforementioned algorithm and its implementation, a newer version was released, C5.0/See5, that is supposed to perform even better[9]. Some of the most noticeable improvements are that it allows a separate cost to be defined for each predicted/actual class pair; if this option is used, the algorithm constructs classifiers to minimize the expected misclassification costs rather than error rates. Furthermore, C5.0 has provision for a case weight attribute that quantifies the importance of each case; if this issue appears, it attempts to minimize the weighted predictive error rate. Some more improvements are that C5.0 is significantly faster than C4.5 (several orders of magnitude) and it is more memory efficient than its predecessor. One particularity that may help the use case we are studying is that C5.0 gets similar results to C4.5 with considerably smaller decision trees and it offers support for boosting, which is supposed to improve the trees and give them more accuracy. Finally, a new and interesting addition is that a new option was implemented and it automatically winnows the attributes to remove those that may be unhelpful.

To explain further, winnow is a technique from machine learning which refers to learning a linear classifier from labeled examples, very similar to the perceptron algorithm. However, Winnow performs much better when many dimensions are irrelevant as it is a simple algorithm that scales well to high-dimensional data. During its training phase, the winnow algorithm is presented with a sequence of positive and negative examples, from which it ultimately learns a decision hyperplane that can then be used to label new examples as positive or negative. Thus, the algorithm is suitable to be used in a setting where the learning and the classification phase are not clearly separated.

Conclusions

Our research has confirmed that Deep Learning techniques and Neural Networks are not suitable for the task and that a classic Machine Learning approach is more beneficial. This is mostly because decision trees seem to perform best in classification tasks of this type.

One of the most prominent setbacks for the actual implementation of a ML-based prediction feature is the absence of a dataset to train our algorithm and fine-tune it in order to be compatible with the data offered by the given application. Furthermore, the attempts to make use of the newest approach for ID3-type algorithms have not been successful. This is most probably caused by the lack of experience in the field.

References

[1] A. Alhassan, B. Zafar, and A. Mueen, *Predict Students' Academic Performance based on their Assessment Grades and Online Activity Data*, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 4, Jun. 2020, https://doi.org/ 10.14569/IJACSA.2020.0110425
[2] M. A. Al-Barrak and M. Al-Razgan, *Predicting Students Final GPA Using Decision Trees: A Case Study*, International Journal of Information and Education Technology, vol. 6, no. 7, pp. 528–533, 2016, https://doi.org/10.7763/IJIET.2016.V6.745.
[3] W. H¨am¨al¨ainen and M. Vinni, *Classifiers for educational data mining*, Handbook of Educational Data Mining, 2010.
[4] D. Garc´ıa-Saiz and M. Zorrilla, *Comparing classification methods for predicting distance students' performance*, The Journal of Machine Learning Research, 2011.
[5] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.
[6] D. Kabakchieva, *Predicting student performance by using data mining methods for classification*, Cybernetics and Information Technologies, vol. 13, 2013
[7] Q. A. Al-Radaideh, A. A. Ananbeh, and E. M. Al-Shawakfa, *A classification model for predicting the suitable study track for school students*, International Journal of Research and Reviews in Applied Sciences, vol. 8, 2001.
[8] A. Nandeshwar and S. Chaudhari. (2009). *Enrollment Prediction Models Using Data Mining*, [Online] Available: http://nandeshwar.info/wp-content/uploads/2008/11/DMWVU_Project.pdf
[9] https://www.rulequest.com/see5-comparison.html
[10] N. Littlestone, *Learning quickly when irrelevant attributes abound*, Machine Learning 2:253–318, 1988, https://www.cs.yale.edu/homes/aspnes/pinewiki/attachments/WinnowAlgorithm/winnow.pdf