Predicting Student Grades - a Comparison of Decision Trees & Naïve Bayes

Judith Grieves

Problem Description and Motivation

- It is useful in education courses to be able to predict student grades in order to inform teachers and allow early intervention, where necessary, to improve performance.
- This work uses the 'Student' dataset, containing demographic, lifestyle and socio-economic data on students from 2 secondary schools, as well as their past performance and final grades in a Portuguese Language course. The dataset was obtained from the UCI machine learning repository [1] and was initially collected by Cortez & Silva [2].
- We use *attribute*s of the dataset as *predictive variables* to train 2 different Machine Learning models to predict the students' Final Grade.
- The experiments will evaluate the performance of Decision Tree (DT) and Naive Bayes (NB) methods and their ability to solve the prediction problem, comparing results and aspects of the modelling process.

Initial Analysis of the Dataset

- The 'student-por' dataset contains 649 *instances*, each with 32 predictive *attributes* and a Final Grade (G3) to be predicted.
- The Final Grade is a numeric *attribute* with values of 1-20 (Fig. 1). Exploratory Data Analysis identified 15 outlier rows with Final Grades of 0. It is assumed these grades were unknown and their inclusion could confound any predictions. They will be removed leaving 634 usable rows.
- To simplify the problem for the machine learning algorithms, we bin the Final Grade numeric values to create a new *binary classifier*: passFail (1/0), a feature set as '1' when Final Grade >= 10, otherwise '0'.
- A *correlation* analysis (Fig 2) shows which features were most closely correlated with the Final Grade. The most significant are: G1, G2 (previous grades) (Fig.3). This analysis will be used in the selection of suitable predictor variables.

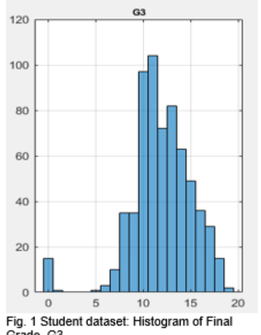

Fig. 1 Student dataset: Histogram of Final Grade, G3.

final grade	predictors	Correlation Coefficient
G3	G2	0.93
G3	G1	0.87
G3	failures	-0.39
G3	higher	0.34
G3	Medu	0.27
G3	studytime	0.25
G3	school	-0.22
G3	Dalc	-0.21
G3	Fedu	0.20
G3	absences	-0.20

Fig. 2 Student dataset: Correlation analysis of predictive features against G3, Final Grade.

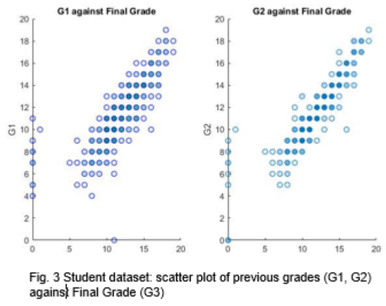

Fig. 3 Student dataset: scatter plot of previous grades (G1, G2) against Final Grade (G3)

Summary of the ML models

A **Decision Tree is** a non-probabilistic, *supervised* learning model, consisting of a tree structure where each node divides the data according to the values of a single feature. The DT algorithms use *information gain,* measuring how well a given *attribute* separates the training examples [5] ,to create the tree nodes.

Advantages

- DT algorithms naturally select the most appropriate features to use and the resulting models are easy to understand and visualise. Algorithms are top-down and relatively fast to run.
- DT models are particularly suitable for problems with a fixed set of *attribute*s and small number of *disjoint* possible values [6].

Disadvantages

- Rule-based classification systems include sharp cut-offs for *attribute*s [6].
- DT can be prone to *overfitting* the training data [5] and their use of a top-down algorithm without any backtracking can generate non-optimal solutions.

A **Naïve Bayes** classifier is a probabilistic supervised learning method using Bayes theorem to calculate the *conditional probabilities* of variables and thereby the most likely classifier value for each test instance.

Advantages

- The NB algorithm is simple and therefore easy to understand.
- Although the NB assumes independence of predictor variables, it produces surprisingly accurate results.

Disadvantages

- NB can suffer from *overfitting*. Feature selection can mitigate this problem.
- Algorithm processing time greater than for DT.

Hypothesis Statement

- Our experimental hypothesis is that the student's Final Grade classification can be predicted to a useful degree of *accuracy*, defined here as 85% or greater.
- We hypothesise that this *accuracy* can be achieved with both machine learning models: Naive Bayes and Decision Tree.
- Given previous research [3] supporting its use in this and other [4] types of problem, it is predicted that a Naive Bayes model will achieve a more accurate result than a Decision Tree.

Training and Evaluation Methodology

MATLAB will be used to create the experiments which will be conducted as follows:

- READ & ANALYSE data : to inform cleaning, wrangling and choice of variables
- CLEAN, WRANGLE & CODE DATA to prepare for the training algorithms
- SPLIT TEST/TRAIN: original dataset is randomly split into test and training sets

For each model and experimental set of (*hyper)parameters* iterate over:

- TRAIN MODEL using MATLAB functions (fitctree / fitcnb) and the training dataset, for each combination of parameter values in a grid search, using *k-fold* error validation.
- TEST MODEL using test dataset to evaluate the ability of the trained network to *generalise*, by making predictions on previously unseen data.
- OUTPUT RESULTS: parameter choice and performance metrics will be output to an xlsx file for analysis.

For each set of experiments:

- EVALUATE RESULTS using training validation errors, *accuracy*, *sensitivity* and *specificity* using *confusion matrices*, training time taken, and ability to *generalise*.
- CHOOSE BEST PARAMETERS for each set of experiments.
- RERUN TRAIN/TEST using best parameters of previous experiments

OUTPUT SAVED MODELs using optimum parameter sets

Parameters and Experimental Results

Grid Search Parameters

Train/Test split %

Feature Selection subsets

Decision Tree: minLeafSize, maxNumSplits, minParentSize

Naïve Bayes: Kernel Smoother Type

Decision Tree Accuracy %				Naive Bayes Accuracy %		
Train	Test	Overall	Predictive Vars	Train	Test	Overall
94.2	93.2	93.7	*grades only*	93.5	93.2	93.4
97.5	91.0	94.3	*all vars*	89.8	88.3	89.1
94.8	81.5	88.1	*all excl. grades*	86.2	82.1	84.1

Fig.4 Output of the final trained models: Decision Tree and Naïve Bayes by Predictor Variable sets.

- As hypothesised, the best of each machine learning models were able to achieve over the target 85% *accuracy* (Fig. 4) when including the highly correlated Previous Grade predictors G1 & G2.
- The best models, in terms of testing accuracy, for both algorithms, were achieved using predictor subset 'Previous Grades only' at 93.2%. In this case, the hypothesis that Naïve Bayes would be the more accurate was not proven as both models achieved identical results. Note that for all available predictors, the DT obtained better results (91% v 88%) but for predictors excluding the previous grades, NB was able to achieve a slightly higher accuracy (82.1% v 81.5%). A test/train split of 50:50 resulted in the least *overfitting*.
- The Decision Tree hyperparameters, used to control tree depth, gave the best performing model for: minLeafSize=18 maxNumSplilts=260 and minParentSize=20 . The best Naïve Bayes model used the 'normal' Kernel Smoother Type.

Analysis and Critical Evaluation of Results

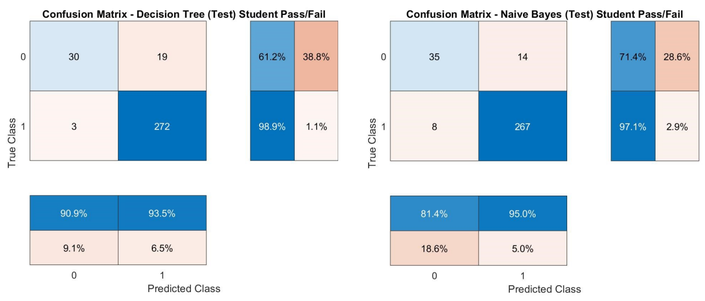

Fig.5 Confusion Matrices of Test Accuracies for (a) Decision Tree (b) Naïve Bayes

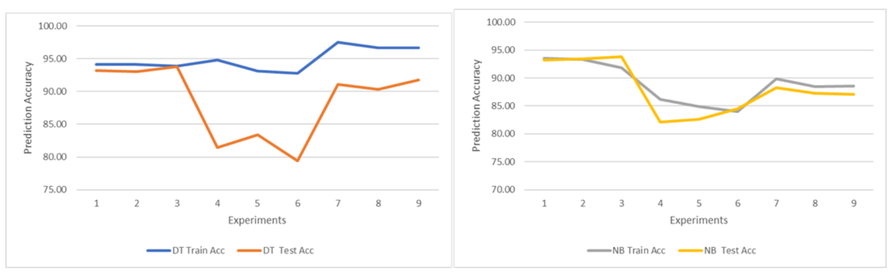

Fig.6 Plot of Train and Test Accuracy per Experiment (a) Decision Tree (b) Naïve Bayes

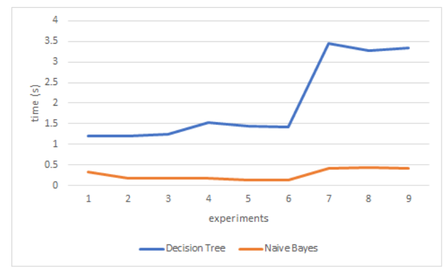

Fig.7 Plot of Model Training Times by Model Type

- The DT model was able to attain over 90% accuracy using all predictor variables, demonstrating that the algorithm could be more effective than NB at choosing the best predictors, regardless of the number of variables submitted to the trainer function. For the NB model, careful selection of the predictor variables (based on the earlier *correlation* analysis) was able to achieve a 5% increase in test *accuracy* over the next best model. That both models were able to show improvement when employing feature selection agreed with the findings of Rahmadani et al [7]. NB was also the least prone to *overfitting* for all predictor subsets. Conversely, for large predictor variable subsets, DT did exhibit overfitting.
- The *confusion matrices* for the models (Fig. 5) show that both had high *sensitivity*, achieving over 95% 'pass' prediction success for train and test datasets. For *specificity*, the NB was able to predict 71% 'fails' correctly whilst DT predicted 61%, so the NB appeared to be more balanced in the type of result it was able to correctly predict. The failures appeared to be more difficult to predict correctly. Further analysis of the underlying dataset shows that these failures occurred at the pass/fail threshold of the classifier and when the student grades declined over the term, rather than steadily improving as seemed to be the case for most instances. This could be because the dataset was not balance over the classifier and there were fewer 'fails' to learn from in the test set.
- We can see more clearly the extent of overfitting for the DT in Fig. 6, with greater discrepancies in accuracy between the train and test results for certain parameter sets. Further tuning of the experiments for these cases could be helpful in identifying any possible mitigating factors. By contrast, the Naïve Bayes had more consistency between the values so we can be more confident that the trained model will also work well on other unseen data. The overview of the NB method highlighted mitigation of overfitting, by pre-selection of good predictor variables, and this was found to be the case (Fig. 4).
- The results of the predictor subset experiments show that inclusion of previous grades as predictors always gave better results than without. However, other predictors, such as the socio-economic indicators, also gave a respectable accuracy of 88% for the Decision Tree. These results are useful for domain stakeholders to highlight students at risk of failure. The findings also accord with the experimental findings in [2] and [3].
- Over all experiments, the time taken to train a model (Fig. 7) was consistently shorter for the NB than the DT. This is not what was expected from previous research. For this Student dataset, timing was not an issue but, for larger or more complex datasets, this result could be more important in choosing a model. Further work should determine whether these results are repeated at scale .
- To choose the optimal DT training *hyperparameters,* grid search results were evaluated (see supplementary material for details). The train/test split was analysed over to evaluate whether the dataset was large and diverse enough to provide a sufficiently balanced split. These experiments resulted in similar distributions of the classifier values (84-5% 'pass') so this was not seen to be a problem. It would be interesting to alter the threshold in order to partition the *binary classifier* so that there are equal numbers of successes and failures and evaluate the impact of this on the model outcomes.

Lessons Learned and Future Work

- The identical results of both methods for the best models may be as a result of the previous grades being such accurate predictors: a good result for a real world problem but possibly not so useful in this comparison. A more detailed analysis of the results for the other predictor subsets could better highlight the pros and cons of the methods.
- For the classifier, a better balanced dataset could give better specificity. This could be done by creating the classifier with a different grade threshold. Binning the Final Grade further as a multi-class classifier ,as used by [2], may also improve results and be valuable to users of this type of data. Future work should repeat the experiments to include this.
- There is a second dataset related to this work, an identically structured set of data for a Mathematics course in the same schools. Analysis of the similarities and differences between the 2 datasets and the effects of merging them could yield useful results showing whether these predictions are generic or specific to subject areas.

[1] https://archive.ics.uci.edu/ml/datasets/Student+Performance

[2] 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.

[3] KOTSIANTIS, S., PIERRAKEAS, C. & PINTELAS, P. 2004, "PREDICTING STUDENTS' PERFORMANCE IN DISTANCE LEARNING USING MACHINE LEARNING TECHNIQUES", Applied artificial intelligence, vol. 18, no. 5, pp. 411-426.

[4] arbosa, R.M., Nacano, L.R., Freitas, R., Batista, B.L. & Barbosa, F. 2014, "The Use of Decision Trees and Naïve Bayes Algorithms and Trace Element Patterns for Controlling the Authenticity of Free-Range-Pastured Hens' Eggs", Journal of food science, vol. 79, no. 9, pp. C1672-C1677.

[6] Gopal, M. 2019, Applied machine learning, McGraw-Hill Education, New York, Chapter 8.

[5] Mitchell, T. 1997, *Machine learning,* [International]. edn, McGraw-Hill, London;New York; pp.52-77.

[7] Rahmadani, S., Dongoran, A., Zarlis, M. & Zakarias 2018, "Comparison of Naive Bayes and Decision Tree on Feature Selection Using Genetic Algorithm for Classification Problem", Journal of physics. Conference series, vol. 978, pp. 012087.