**CSC105M Final Project**

Ryan Austin Fernandez

CS-ST Student

2401 Taft Avenue,

Manila, Philippines 1004

(+63)917-621-6469

ryan\_fernandez@dlsu.edu.ph

Clarisse Felicia Poblete

CS-ST Student

2401 Taft Avenue,

Manila, Philippines 1004

(+63)915-624-5417

clarisse\_poblete@dlsu.edu.ph

**ABSTRACT**

A student’s lifestyle affects their academic performance. In an attempt to find out which factors have the largest effect on the academic performance of a student, machine learning techniques such as multilinear regression, decision trees, and neural networks were used on the alcohol consumption dataset from UCI Machine Learning Repository to predict if a student would pass or fail, which is important in helping students before they fail by possibly using these models in predicting their status. Bagging was the ensemble method used for this study. Multilinear regression was deemed unsuitable for this dataset due to the low correlations of the variables; decision trees and neural networks produced better results, with <results here> <point out which factors affect the most> <point out the most effective method and the most effective rules> <recommendations>

**I. Dataset Description**

A student’s lifestyle often has a large impact on their academic performance. This includes what school they are attending, their age, where they live, how large their family is, their financial status, how much they study, how much they drink, and many more factors to list here. A study by Amran, H. & Pagnotta, F. (2016) has already been done on the Student Alcohol Consumption Data Set that has been collected for this project. In their study, they use lifestyle factors to predict whether a student is an alcoholic or not using decision trees in order to help predict if future students would succumb to alcohol. They were successful in doing so, garnering an accuracy of almost 92%.

Data collection was done by downloading the dataset from the UCI Machine Learning Repository. The downloaded file was a zip file which contained two (2) comma-separated values (csv) files student-mat.csv and student-por.csv which contained the information as enumerated in Table 1 for math and Portuguese respectively.

**Table 1. Attributes of the Student Alcohol Consumption Data Set**

|  |  |  |
| --- | --- | --- |
| **Attr #** | **Attribute** | **Description** |
| 1 | school | student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira) |
| 2 | sex | student's sex (binary: 'F' - female or 'M' - male) |
| 3 | age | student's age (numeric: from 15 to 22) |
| 4 | address | student's home address type (binary: 'U' - urban or 'R' - rural) |
| 5 | famsize | family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3) |
| 6 | Pstatus | parent's cohabitation status (binary: 'T' - living together or 'A' - apart) |
| 7 | Medu | mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 5th to 9th grade, 3 secondary education or 4 higher education) |
| 8 | Fedu | father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 5th to 9th grade, 3 secondary education or 4 higher education) |
| 9 | Mjob | mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at\_home' or 'other') |
| 10 | Fjob | father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at\_home' or 'other') |
| 11 | reason | reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other') |
| 12 | guardian | student's guardian (nominal: 'mother', 'father' or 'other') |
| 13 | traveltime | home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour) |
| 14 | studytime | weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours) |
| 15 | failures | number of past class failures (numeric: n if 1<=n<3, else 4) |
| 16 | schoolsup | extra educational support (binary: yes or no) |
| 17 | famsup | family educational support (binary: yes or no) |
| 18 | paid | extra paid classes within the course subject (Math or Portuguese) (binary: yes or no) |
| 19 | activities | extra-curricular activities (binary: yes or no) |
| 20 | nursery | attended nursery school (binary: yes or no) |
| 21 | higher | wants to take higher education (binary: yes or no) |
| 22 | internet | Internet access at home (binary: yes or no) |
| 23 | romantic | with a romantic relationship (binary: yes or no) |
| 24 | famrel | quality of family relationships (numeric: from 1 - very bad to 5 - excellent) |
| 25 | freetime | free time after school (numeric: from 1 - very low to 5 - very high) |
| 26 | gout | going out with friends (numeric: from 1 - very low to 5 - very high) |
| 27 | Dalc | workday alcohol consumption (numeric: from 1 - very low to 5 - very high) |
| 28 | Walc | weekend alcohol consumption (numeric: from 1 - very low to 5 - very high) |
| 29 | health | current health status (numeric: from 1 - very bad to 5 - very good) |
| 30 | absences | number of school absences (numeric: from 0 to 93) |
| 31 | G1 | first period grade (numeric: from 0 to 20) |
| 31 | G2 | second period grade (numeric: from 0 to 20) |
| 32 | G3 | final grade (numeric: from 0 to 20, output target) |

Since building a model by combining these two files was not possible since the identities were removed, data selection was performed by selecting the larger dataset, which was the dataset on the Portuguese language class, which contained six hundred and fifty (650) instances compared to the Math class’ three hundred and ninety-six (396).

**2. Data Preprocessing**

The dataset had no missing values. All the attributes were discrete. As such, data cleaning and transformation was not too drastic. Since regression and neural networks were two of the machine learning techniques that were being considered, some normalization was necessary since most of the attributes were discrete in nature.

For binary attributes, yes was converted to 1 and no was converted to 0. For nominal attributes with two classifications, one class was assigned 1 and another was assigned 0. This was done for attributes 1, 2, 4, 5, 6, 16, 17, 18, 19, 20, 21, 22, and 23.

For nominal attributes with three or more possible values, the attribute was split into multiple new attributes equal to *n – 1* with n being the number of possible attributes. For example, the guardian attribute can have the value “mother”, “father”, or “other”. Since there are three classifications, there will be two resulting attributes: guardM and guardF. If the value is “mother”, guardM will be 1, guardF will be 0. If the value is “father”, guardF will be 1, guardM will be 0. If the value is “other”, both guardF and guardM will be 0. This transformation was done for attributes 9, 10, 11, and 12.

For ordinal variables, number values were assigned based on their hierarchy. In the dataset, this means attributes 7 and 8, which were already given numerical values so these were retained.

After these transformations, all attributes were now numerical, but not all were in the range [0,1]. These attributes were retained as is for the decision tree model. For the neural network and regression models, however, these were normalized based on the maximum and minimum values per attribute using the formula

xnew = (xold – xmin) / (xmax – xmin) \* (xnewmax – xnewmin) + xnewmin

where Xnewmax and Xnewmin were 1 and 0 respectively. This transformed all columns to the range [0,1].

Finally, since decision trees are classifiers, the target variable, the final grade, was classified into Pass or Fail. 12 above is a Pass while anything below is a Fail.

**3. Feature Selection**

**3.1. Multilinear Regression**

In building a multilinear regression model, the pairwise Pearson correlation coefficients were taken to see if a multilinear regression model was appropriate.

To do this, all the data was stored in a two dimensional table *D* with *n* rows and *a columns*, where *n* is the number of instances and *a* is the number of attributes. *Dij* then contains the ith instance’s jth attribute value.

A product table *P* was constructed, which was an *a x a* table that contained the sum of the pairwise products of each attribute for all instances, which is to say

A separate array *S* was made, containing *a* values, which is the sum for each attribute.

Finally, the correlation matrix *C* was computed. Using the Pearson correlation formula

This means that to compute each value of the matrix *C*, the following formula was used.

After computing for all the pairwise correlations, a threshold of 0.8 for independent – dependent pairs and a threshold of 0.6 for independent – independent pairs was used to determine if the two were highly correlated. Unfortunately, there were barely any pairs that reached these thresholds. Attributes 31 and 32 were one pair that were highly correlated, but this was irrelevant since the first two period’s grades would obviously influence the final grade. Another high correlation was the attributes that resulted from attribute 12, guardM and guardF, which were also irrelevant since it is obvious that if the mother does not guard a child, the father would.

Based from these results, the multilinear regression model was deemed unsuitable for this dataset.

**3.2. Decision Trees**

The next machine learning technique that was used was decision trees. Since the C4.5 algorithm determines which features give the most information gain, all features were initially input into the algorithm. Only the features found in the final decision tree were included.

**3.3. Neural Networks**

The final machine learning technique used was neural networks. Since neural networks are robust when it comes to noise in the data, all features were included in the training of the neural networks and the corresponding weights of the neurons would discriminate which features were useful.

**4. Visualization**

**5. Analytics**

This project aims to predict a different variable from another variable from Amran, H. & Pagnotta, F. (2016) who predicted the alcohol intake. This project aims to predict the final grade in the class, or more specifically, if the student will pass or fail. This study aims to see which variables affect the final grade of the student and how much.

To do this, three machine learning techniques are to be considered: multilinear regression, decision trees, and neural networks.

Multilinear regression produces a function that maps multiple independent variables to a single dependent variable. To do this, pairwise correlations must first be taken between the attributes to determine which attributes are highly correlated with the target variable. Once these attributes are found, the highly correlated independent attributes must then be pruned to reduce noise in the data. Multilinear regression is not effective if the independent and dependent variables are not highly correlated. As mentioned in section 3, this method was deemed not feasible due to the low correlation of the independent and dependent variables.

Decision trees decomposes the data into a number of if-then rules. This is done by considering the gain of information on each variable, which is the amount of uncertainty reduced by splitting the dataset on the various values of that variable, and always splitting on the attribute that results in the maximum gain. This is the ID3 algorithm, which is not very robust when it comes to avoiding overfitting the data and continuous variables, which is compensated by the C4.5 algorithm. WEKA, a machine learning tool, has implemented the C4.5 algorithm as the J48 algorithm (Mitchell, M., 1997).

The final machine learning technique that will be considered is neural networks, which simulate the neurons in the human brain. By getting the linear combination of the input variables with a set of weights and using an activation function to fire each neuron, output values are produced, which may be interpreted as classification, by determining which output has the highest value, or by regression, by reversing any normalization done on the output (Mitchell, M., 1997).

**5.1. Decision Trees**

For this method and the next, neural networks, an ensemble method, bootstrap aggregating or bagging was used. 80% of the instances was considered for the bootstrap size. Bootstraps were generated using random selection with replacement. Five bootstraps were generated.

In generating the decision trees, WEKA was used. Each bootstrap was fed to the tool to produce a unique decision tree using the J48 algorithm, which is WEKA’s implementation of the C4.5 algorithm. The resultant trees were very accurate within their bootstrap. Their accuracies were 95.183%, 96.9171%,96.1464%,96.1464%,and 97.3025%.

Since there was no way to export the trees in any way, Java was used in implementing a text parsing program for the WEKA-produced decision trees. The five trees were then run on the dataset, each tree voting for the classification of Pass or Fail, majority being the final decision.

Moreover, the frequency for each leaf that was reached and led to a correct classification was recorded to note the most useful rules.

**5.2. Neural Networks**

A custom implementation of neural networks was created for exploratory purposes. Initially, a slight implementation error where the backpropagation was being performed at the wrong time occurred, which led to the model being scrapped, but after debugging, the model worked.

The same bootstraps were used for neural network training. The network had three layers: thirty-nine (39) input neurons for the input layer, twenty (20) sigmoid neurons for the hidden layer, and two (2) sigmoid neurons for the output layer. If the first output neuron produced a higher value than the second, the student was classified as Fail; otherwise, the student was classified as Pass.

For output representation, a pass would have the output neurons targeting a 0.1 and a 0.9 while a fail had them targeting a 0.9 and a 0.1. These values were chosen since an extremely high absolute value for the linear combination would be necessary to achieve a 0 or a 1 in the sigmoid activation functions while 0.1 and 0.9 were attainable.

The neural networks were trained with a learning rate of 0.2 and a momentum of 0.3. Initial weights were randomized between -0.1 and 0.1. The termination condition was if the mean squared error (MSE) was below 0.04 or 300 epochs has been reached.

The accuracies of the five neural networks within their own bootstraps were 100%, 100%, 99.8077%, 99.8077%, and 99.5192%.

The five neural networks were then run on the full dataset, with the majority decision being considered.

A regressor using neural networks was also attempted by normalizing the grades from 0.1 to 0.9, but the performance within the individual bootstraps was not desirable, with a 24.3548% accuracy, so a full model was not trained.

**6. Interpretations, Findings, and Conclusions**

The decision tree forest performed well on the full dataset. The classification accuracy was 89.9846%; the classification error was 10.0154%; the sensitivity (positives correctly classified) was 95.6897%; the specificity (negatives correctly classified) was 83.3887%. Table 6.1. shows the most successful rules.

**Table 6.1. – Most Frequently Correct Rules**

|  |  |  |
| --- | --- | --- |
| Correct Predictions | Wrong Predictions | Rule |
| 158 | 31 | failures = 0 ^ higher = yes ^ Mjob != home ^ Walc <= 3 ^ schoolsup = no ^ school = GP ^ internet = yes ^ age <= 18 -> Pass |
| 139 | 24 | higher = yes ^ failures = 0 ^ school = GP ^ nursery = yes ^ internet = yes ^ schoolsup = no ^ Dalc <= 1 -> Pass |
| 110 | 24 | failures = 0 ^ higher = yes ^ Mjob != home ^ Dalc <= 2 ^ Fjob != teach ^ absences <= 3 ^ health <= 4 -> Pass |
| 88 | 5 | failures > 0 ^ age <= 19 -> Fail |
| 88 | 3 | failures > 0 ^ Medu <= 3 ^ Fedu > 0 -> Fail |
| 81 | 9 | failures = 0 ^ higher = yes ^ school = GP ^ schoolsup = no ^ internet = yes ^ nursery = yes ^ Fedu <= 3 ^ Dalc <= 1 ^ Fedu > 1 -> Pass |
| 67 | 2 | failures > 0 ^ Medu <= 2 -> Fail |
| 64 | 7 | failures = 0 ^ higher = yes ^ Mjob != home ^ internet = yes ^ absences <= 8 ^ schoolsup = no ^ studytime > 2 -> Pass |
| 63 | 2 | failures > 0 ^ Walc > 1 -> Fail |
| 61 | 9 | failures = 0 ^ higher = yes ^ Mjob != home ^ internet = yes ^ absences <= 8 ^ schoolsup = no ^ studytime <= 2 ^ school = GP ^ Medu > 2 ^ Mjob != teach -> Pass |

The variables that commonly resulted in the highest information gain when producing the trees within each booststrap

The neural networks performed much better. Using the five neural networks trained using bagging, the classification accuracy for the original dataset from where the bootstraps were generated was 92.9122%; the classification error was 7.0878%; the sensitivity was 94.5402%; the specificity was 91.0299%.

It is, however, noticeable that the performance of the individual networks within their bootstraps is better than the ensemble network with the entire dataset. This may have been caused by overfitting of the data from the training in the individual bootstraps.

<comparison>

<conclusion>

<recommendations>

**References**

Amran, H. & Pagnotta, F. (2016). Using Data Mining to Predict Secondary School Alcohol Consumption. *University of Camerino.* doi: 10.13140/RG.2.1.1465.8328

Baesens, B. (2014). *Analytics in a Big Data World: The Essential Guide to Data Science and its Applications.* NJ: John Wiley & Sons.

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

Mitchell, T. (1997). *Machine learning*. McGraw-Hill.

Stockburger, D.W. (n.d.) Multiple Regression With Categorical Variables. Retrieved July 27, 2016, from Psychological Statistics at Missouri State University: http://www.psychstat.missouristate.edu/multibook/mlt08m.html