# Report on Data Mining project: Video Game Value of Trading

November 30, 2019

Russell Snelgrove

Corbin Graham

SENG 474

University of Victoria

**Introduction**

Many online video games have an in game marketplace where players can trade in game items for in game currency. This in game trading of items has a tendency to mimic real world markets with ideas like supply and demand and world news affecting market values. Old School RuneScape is a massive multiplayer online role playing game, released in 2013 [1], where players can participate in trading items they acquire for in game currency or vice versa. The in game currency, gold pieces or gp for short, are most commonly spent and gained at the central trading hub called the Grand Exchange (GE). In the GE, players are able to place buying offers on items in varying quantities at a given price. Additionally, since the GE is fully player driven, you can sell items on the Grand Exchange for a reserve price set by the player. When putting in offers for a particular item the game defaults to a suggested price. This default suggested price is an average price for that item on that day. Runescape's GE is an entity worthy of closer analysis and we will do this in three ways.

The first aspect to look at involves an item called a "Nature rune". Players are able to use a "Nature rune" to turn items into a set amount of gold pieces. This process is commonly referred to as "high alching" an item. If done correctly, players are able to make a profit by purchasing items from the GE and high alching them. In order to do this, the player must ensure that the cost of the item from the GE plus the cost of the "nature rune" from the GE is less than the high alching value. We would like to see if it is possible to predict if an item will be profitable without knowing the current value of a "Nature rune". We will do this using the J48 algorithm.

The second aspect will look at the profitability of the "high alching" process over time. This process will use linear regression as a supporting model.

The final aspect of interest involves the trade volume of an item and if it can be used to predict if an items daily trading value average is above or below its six month trending value. This is useful for the process known as "flipping". "Flipping" is the action of buying an item below market value and attempting to sell it above market value. We would like to see, using the J48 algorithm, if the trade volume alone is sufficient for checking profitability.

The gathered data set includes 25 independent items with each item containing 180 days worth of data. Each day includes the daily average trade value, six month trend value, and daily trade volume.

The motivation behind this report is initially based off personal interest. Having been active players in the past, we wanted to more closely analyse the many players ability to make money in game by buying items at a lower price and selling them at a higher price. We would like to see if it's possible to predict profitability, or the value of an item, based on data that we collect.

The approach to this project is done in multiple steps. The first step is gathering the data via web scraping Old School Runescape's GE pages and processing the raw HTML with the Ruby programming language. Excel will be used to generate graphs to create visual representations of the gathered data. Weka will then be used to run some of the data sets through the J48 algorithm.

**Dataset**

The datasets for this project are generated from a Ruby script that collects Old School Runescape's raw HTML pages [2]. The Ruby script will then scrape the raw HTML for data relating to trades and daily averages over the last six months. Data is finally written into csv files so that they can easily be processed through microsoft excel or google spreadsheets. Data related to trades involved with tradable items in Old School RuneScape is available dating up to six months prior to the current date.

The items that have been selected for analysis are items that have relative stability in the game. As the game continues to develop, new items are added into the game that have unpredictable effects on the games economy. The selected items for this report have been in the game long enough that their prices are relatively stable and players understand the uses of the item.

The data was collected from the Old School RuneScape Grand Exchange website [2]. The scraped data came from embedded javascript that was being used to display graphical representations of the collected data. The method for gathering the data from the website was to perform an http *get request* using the *net/http* ruby gem [3]. This *get request* was repeated for each one of the 25 items. The script was written to loop through a list of web addresses and make a rested *get request* for each addresses raw HTML. This raw HTML is then scraped through to gather the data that is required for the analysis. Once the data has been sanitized, it is pushed into a CSV.

Each item has its own CSVfile along with a larger file that contains all item data concatenated together. Each of the items CSV's contain six months of data that tracks the date, the daily traded average price, the six month trend leading up to that date, and the total amount of that item traded on the given date. The files have four columns and 180 rows. In addition, some files will be manipulated to contain a true or false value for whether the daily traded value is greater than the six month trend leading up to that date.

**Preprocessing**

The preprocessing portion of this project involves the scraping of HTML, preparation of CSVs, and the preparation for analysis. The HTML is gathered as discussed in the Dataset section of this report by the ruby script that allows us to gather raw HTML with embedded values of prices for items. Data that was collected was complete with no missing values. If there was an event of

missing values, the data for that range of time would be ignored and the analysis would pertain to only the data present.

All HTML from the GE website has a standard format that allowed for a standard form of parsing and formatting of CSVs. For parsing, the HTML was split on every newline character and scanned using a regex to find all the lines of interest. For each item there was data that populates graphs for one month, three months and six months of data. Parsing was to select only data that pertains to the six months of data that will be focused on. The data was grouped by dates and written into CSVs in chronological order.

In cases of attempting to predict prices, we have decided to split the data into a training set and a validation set. The training set of data will be used to form a prediction for what prices could be in the future. The date range for the training set will be the first 160 days of data for the linear regression model and the first 144 (80%) days for J48 modeling. The validation portion of data will be the most recent 20 days of data for linear regression, and the remaining 36 days for J48. In each case, the validation will be used to see how accurate the algorithms were. Splitting data into these separate groups will only occur for cases where predictions are made and the accuracy of these predictions should be tested.

**Experiment/Analysis**

The experiment for this report is based around pattern recognition and predictability. The J48 algorithm will be used to test if it is predictable to determine if an item is profitable for “high alching” based on its daily average price without the cost of the “Nature rune”. Linear regression will then be used to predict the profitability of high alching an item over time. Finally, J48 will be used to analyze if the volume of an item traded can predict if its daily traded value will be greater than its six month trend average.

# High Alch Profitability

When examining the profitability of high alching based solely on the cost of the main item, it was decided to use J48. In this case was to examine if the item is profitable to high alch based on its daily average price without taking into account the cost of a “Nature rune”. To do this, we used Excel to find the total cost of a “Nature Rune” plus a “high alchable” item to find the total cost of performing a “high alch” on the item. We then created another column that evaluates if this price is less than the total gold received from performing a “high alch”. If it is profitable, the boolean expression of true was given. After this, we put the boolean values into a new file with only the price of the item and formatted it into an arff file. We then decided to see if using the J48 algorithm, which is an extension of ID3[4], would be sufficient for determining if an item was profitable based solely on the cost of the item. From the data that we gathered we saw that this

was only accomplishable with the data for item number 1339. Other gathered item data looked to always be profitable or operating at a loss when “high alching”.

Using Weka, we ran the J48 algorithm with an 80 percent split on the data. The split in Weka allows Weka to use the first 80 percent of the data we provided for training data and the last 20 percent as testing data. Running this resulted in 88.9 percent accuracy for classifying the testing data. The full Weka output can be seen in Appendix B, Figure 3. This level of accuracy was higher than expected. After examining the changes in price, in comparison to more expensive items, item number 1339 is less volatile and has smaller jumps in the daily average prices. This may account for the higher than anticipated accuracy.

Along with examining if an item is profitable based solely on the cost of the item, we also decided to see if we can determine the percentage of profit based on an item that is always profitable without taking into account the cost of a nature rune. The preparation of data for this was almost exactly the same as when examining if an item is profitable based solely on the cost of the item. First we wanted to see if an object could be estimated with high accuracy to be less than one percent profitable to High Alch. We decided to use all the data that was originally collected when scrapping the raw html. Using Weka we saw that we could determine if High Alching would be below one percent with 94.4 percent accuracy. The full results can be seen in Appendix B, Figure 4. Knowing that we can determine when the profit is below 1% we can us that data and inverse it to know when the profit is above 1%. Seeing that the daily average price was chosen as the root makes sense. The other aspect that we were interested in analyzing was to see if we could determine if the percentage of profitability was above 0.75 percent. We selected only daily average and amount traded to determine this percentage of profit. The full results of this analysis can be seen in Appendix B, Figure 5. It should be noted that in analysis as well, the root is daily average. When examining both cases and comparing, when players are wanting to maximize their in game profits, they need to first pay attention to the daily average price to determine whether it is a good time to invest in these items or if they should wait.

## Linear Regression

We decided to use the linear regression algorithm to predict the profitability of “high alching” an item as a function of time. We used a series of different items and took into account the cost of a “Nature Rune”. As an example we will be talking about item number 1185. Using Excel, we found the combined total of the daily average price of the item and the cost of a “Nature rune” for the given date. We then took that total cost and subtracted it from the “high alching” value of the item. If the item was bought on that date for the given daily average price, we can see the amount of profit that a player would receive. With these calculated profits, we were able to graph a chart, as seen in figure 1. Using the first 160 days as training data, we created a prediction line that estimated the trend the profit would continue in. In figure 1 the first 160 days are represented with blue points which is used as the training data. The validation data that is used is the red points, which is the final twenty days of data that were gathered. The prediction line uses only the blue points as contributing data. It is visible that the prediction line has a

steady positive slope. It can be seen that there are a few outliers that do not perfectly fit the model. These individual errors can increase error in extreme cases, however based on the number of data points provided, the few outliers are assigned a smaller weight and can be mostly negated. However looking at the density of the validation data points, we can see that the prediction line was fairly accurate. It would have been nice to use a larger set of data to see if this regression is completely historically accurate.

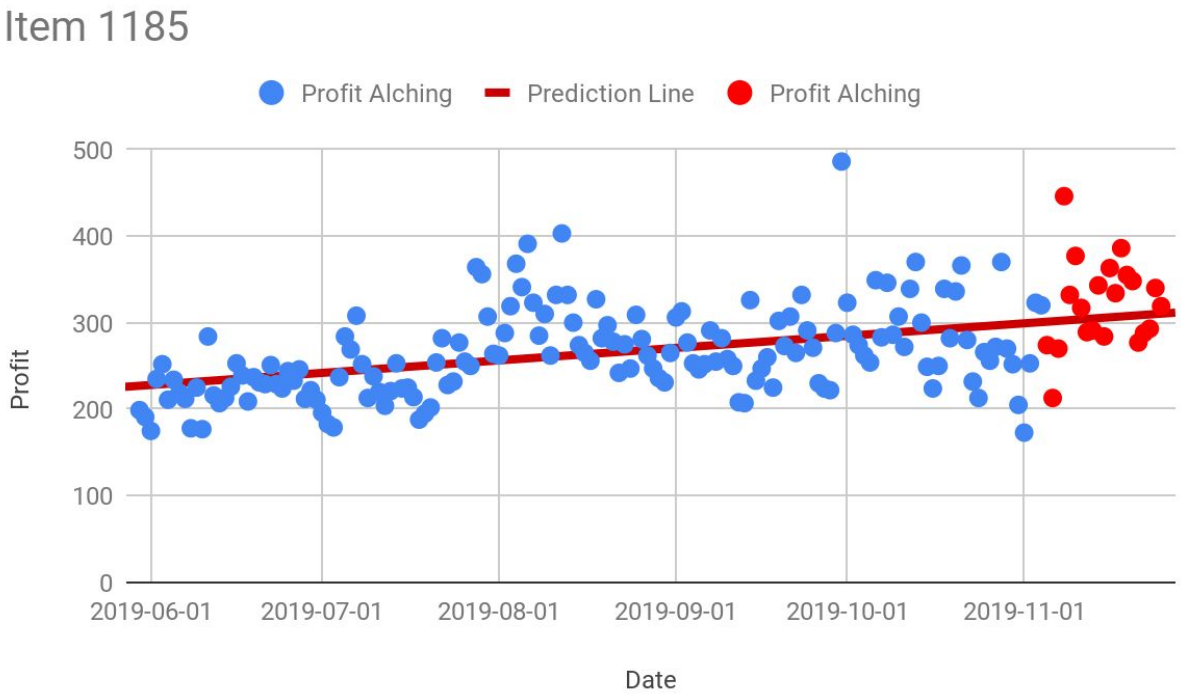
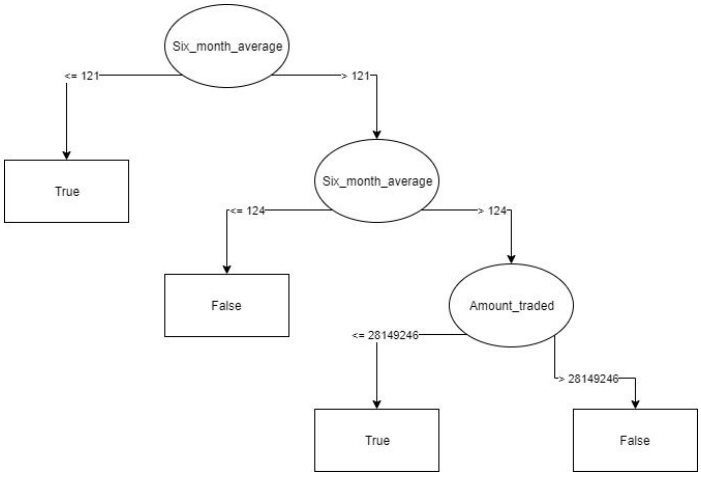



Figure 1: Item 1185 High Alching Profit Linear Regression model

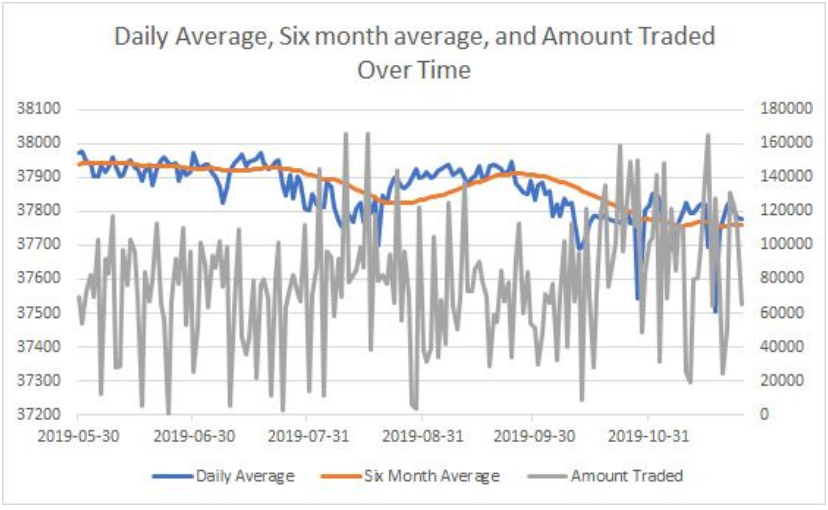
## Prediction of Daily Average Being Greater than Trend based on Amount Traded

The final experiment was to examine if J48 could predict if the daily average price of an item was more or less than the six month trending average based on the daily amount traded. To do this, we used 4 columns: six month trending average, daily amount traded, a boolean checking if the daily amount traded was greater than average, and a boolean checking of the daily average was greater than the six month average. We used the first 80 percent of data for

training and the last 20 percent for validation.




After running the J48 algorithm in weka, the tree seen in figure 2 was outputted.(I uploaded the wrong tree PNG, will fix once taken down for final pass). This tree gave an accuracy of ~80 percent by correctly classifying 29 of the 36 instances. The full results of the weka output can be found in appendix B. These results were somewhat surprising given the relative instability of daily amount traded. Figure 3 shows how intense the fluctuations of amount traded are relative to both the daily average and the six month average of an item. As an optimization for this problem, it may be worth finding smoother amount traded data. Additionally, a larger data set could improve the overall accuracy of this experiment.




**Conclusion:**

Throughout the examination of High Alch profitability, we found the most reliable metric for measuring if an item was profitable, and by how much, was to utilize and understand the daily average price of the item. Through our testing, we found that linear regression worked very well to see the magnitude that this method of prediction can be accurate. Given more data, it would be good to further confirm the direction of our linear prediction. If it were found to be accurate, it would be useful for quickly discovering profitable items over time.

Our claim of the ability to predict the daily average being greater that the six month average based on the amount traded created better than expected results. Given the rather small data set, 180 rows of a single item, the 80 percent accuracy was not anticipated. Repeating this experiment would benefit greatly from a much larger data set, that is, ~1000 items of 180 rows or 2 years worth of data for ~10-20 items.

There are a number of things that could be done to improve the estimation of prices and understanding the historical prices of items. As new items are added into the game, it may take over the niche role of other items, devaluing existing items. The reverse could also be true, if an item is added into the game that requires another already existing item to upgrade it, the demand for the items could increase driving prices upward. If we were to repeat this test in the future, utilization of data mining techniques to measure what items could be affected by upcoming or new updates could help predict prices of items.

**References**

[1] https://en.wikipedia.org/wiki/Old_School_RuneScape

[2] http://services.runescape.com/m=itemdb_oldschool/

[3] https://ruby-doc.org/stdlib-2.6.5/libdoc/net/http/rdoc/Net/HTTP.html

[4] https://en.wikipedia.org/wiki/C4.5_algorithm

[] http://www.eecs.harvard.edu/~cat/vetr.pdf

**Appendix A:**

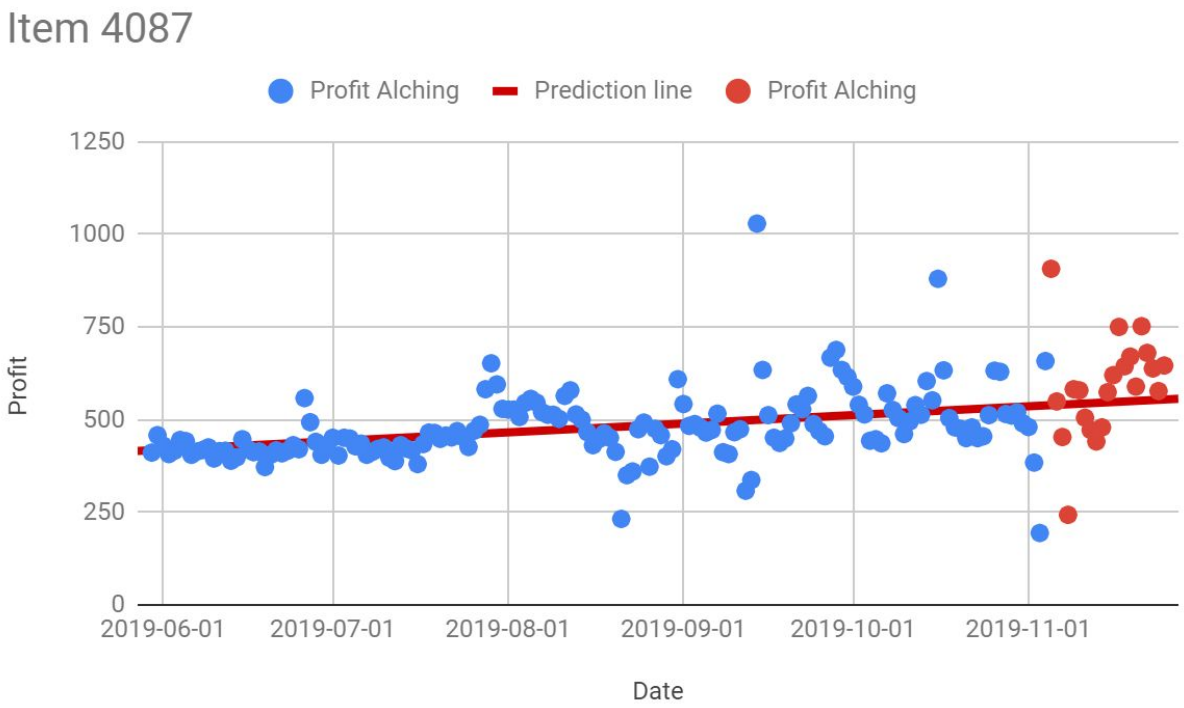



Figure 2: Item 4087 High Alching Profit Linear Regression model

**Appendix B:**

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     osrs
Instances:    180
Attributes:   2
              DailyAverage
              Profitable
Test mode:    split 80.0% train, remainder test

=== Classifier model (full training set) ===

J48 pruned tree
------------------

DailyAverage <= 172: TRUE (115.0/16.0)
DailyAverage > 172: FALSE (65.0/6.0)

Number of Leaves  :     2

Size of the tree :      3


Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          32               88.8889 %
Incorrectly Classified Instances         4               11.1111 %
Kappa statistic                          0.75
Mean absolute error                      0.2083
Root mean squared error                  0.3301
Relative absolute error                 43.4524 %
Root relative squared error             68.3413 %
Total Number of Instances               36

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.917    0.167    0.917      0.917    0.917      0.750    0.875     0.896     TRUE
                 0.833    0.083    0.833      0.833    0.833      0.750    0.875     0.750     FALSE
Weighted Avg.    0.889    0.139    0.889      0.889    0.889      0.750    0.875     0.847

=== Confusion Matrix ===

  a  b   <-- classified as
 22  2 |  a = TRUE
  2 10 |  b = FALSE
```

Figure 3: Item 1339 Weka Results for Alch profit prediction based on item price

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     osrs-weka.filters.unsupervised.attribute.Remove-R4-6
Instances:    180
Attributes:   5
              DailyAverage
              Six_Month_Average
              Amount_Traded
              Profit_greater_than_0.75_percent
              Profit_less_than_1_percent
Test mode:    split 80.0% train, remainder test

=== Classifier model (full training set) ===

J48 pruned tree
------------------

DailyAverage <= 37813
|   DailyAverage <= 37783: FALSE (32.0)
|   DailyAverage > 37783
|   |   Six_Month_Average <= 37842: TRUE (9.0/1.0)
|   |   Six_Month_Average > 37842
|   |   |   DailyAverage <= 37788: TRUE (4.0/1.0)
|   |   |   DailyAverage > 37788: FALSE (7.0)
DailyAverage > 37813: TRUE (128.0)

Number of Leaves  :     5

Size of the tree :      9


Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          34               94.4444 %
Incorrectly Classified Instances         2                5.5556 %
Kappa statistic                          0.84
Mean absolute error                      0.0556
Root mean squared error                  0.2357
Relative absolute error                 15.304  %
Root relative squared error             54.3499 %
Total Number of Instances               36

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.222    0.931      1.000    0.964      0.851    0.889     0.931     TRUE
                 0.778    0.000    1.000      0.778    0.875      0.851    0.889     0.833     FALSE
Weighted Avg.    0.944    0.167    0.948      0.944    0.942      0.851    0.889     0.907

=== Confusion Matrix ===

  a  b   <-- classified as
 27  0 |  a = TRUE
  2  7 |  b = FALSE
```

Figure 4: Item 1319 Weka Results for Predicting less than 1% profitability from High Alching

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     osrs-weka.filters.unsupervised.attribute.Remove-R4-6
Instances:    180
Attributes:   5
              DailyAverage
              Six_Month_Average
              Amount_Traded
              Profit_greater_than_0.75_percent
              Profit_greater_than_1_percent
Test mode:    split 80.0% train, remainder test

=== Classifier model (full training set) ===

J48 pruned tree
------------------

DailyAverage <= 37813
|   DailyAverage <= 37783: FALSE (32.0)
|   DailyAverage > 37783
|   |   Six_Month_Average <= 37842: TRUE (9.0/1.0)
|   |   Six_Month_Average > 37842
|   |   |   DailyAverage <= 37788: TRUE (4.0/1.0)
|   |   |   DailyAverage > 37788: FALSE (7.0)
DailyAverage > 37813: TRUE (128.0)

Number of Leaves  :     5

Size of the tree :      9


Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          34               94.4444 %
Incorrectly Classified Instances         2                5.5556 %
Kappa statistic                          0.84
Mean absolute error                      0.0556
Root mean squared error                  0.2357
Relative absolute error                 15.304  %
Root relative squared error             54.3499 %
Total Number of Instances               36

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.222    0.931      1.000    0.964      0.851    0.889     0.931     TRUE
                 0.778    0.000    1.000      0.778    0.875      0.851    0.889     0.833     FALSE
Weighted Avg.    0.944    0.167    0.948      0.944    0.942      0.851    0.889     0.907

=== Confusion Matrix ===

  a  b   <-- classified as
 27  0 |  a = TRUE
  2  7 |  b = FALSE
```

Figure 5: Item 1319 Weka Results for Predicting greater than 0.75% profitability from High Alching