DATA MINING ASSIGMENT

BRS00035

Contents

Introduction

The project I have been given by Ivor Buquetlowd an owner of over 100 shops who wants to find out why some of his stores are doing better than others. He has asked me to use weka software to build a predictor that will predict the estimated revenue which a store should generate. He has given me data about each of the stores. Using the data provided I can make use of machine learning and data mining techniques to predict a stores revenue.

He has also tasked me with finding out what data is useful and what data is not since collecting the data is rather expensive particularly the population data so will check if the population data is useful in the help of predicting the stores revenues as well as finding out which variables are the most useful and which can be removed from the data. I will also investigate the multilayer perceptron and decision tress are useful as he states that his IT director has some experience of these techniques.

Using machine learning we can make use of many different types of algorithms to help us make very accurate predictions very quickly and efficiently.

I have been asked to use WEKA to find out the prediction. Weka software can use a wide range of data mining algorithms to create prediction models and can be used to analysis the data. WEKA is useful as it can perform these algorithms very quickly, must faster than a human would take so it's a very efficient tool to deal with this problem.

Data Summary

I was given data related to the project which contains 20 attributes:

Town, Country, Store ID, Manager Name, Staff, Floor Space, Window, Car Park, Demographic Score, Location, 40/30/20 min populations (each separated) store age, clearance space, competition number, competition score, Profit and Performance.

Store ID is a numeric unique value given to each of the stores, this isn't very useful for datamining as it doesn't hold any meaning similar to a name and won't give any meaningful data similar to a name and since it's a numeric value it might add to misleading data so it will not be used.

Manger name as like store ID it's the name of a manager at a store which won't make any difference to the datamining process and will be removed.

Country this is only valued as UK this isn't very useful since they are all the same value if we were doing a prediction including other countries with a large set of data for each it would make a difference to see if profits were higher in other countries but since we are only looking at the UK it's not needed and won't impact the overall results so it will be removed.

Town each have it's only unique value this isn't very useful as they isn't enough data in each town to make any difference in the data mining process so it will be removed and since we have the location data which has large set of data for each location we can use that instead.

Performance is a nominal data type showing the relationship between the profit and a performance score ranging from poor, reasonable, good and excellent.

Car Park is a nominal datatype showing if the store has a car park or not with “Yes” or “No” values respectively

Staff is a numeric type showing how many staff is employed at each of the stores

Competition Score is a numeric data type that shows the score of a competitor store

Competition Number is a numeric data type that shows how many competitor stores are near by

Location of store is a nominal data type showing the location of the store raging from values from retail park, shopping centre and high street

the demographic score is a numeric data type showing the score of the demographic such as the targeted cliental

Window is a numeric type showing the size of the windows

Floor space is a numeric data type showing how much floor space is available

Store Age is a numeric data type showing how long the store has been open in years

10/20/30/40 min population is a numeric data type showing the size of the population based on travel distance

Clearance space is a numeric data type showing how much clearance space is in each store

Data Selection

The variables I will be investigating are:

- Performance
- Profit
- Competition Score
- Store Age
- Competition Number
- Min Population’s (40,30,20,10)
- Location
- Staff
- Demographic Score
- Window
- Floor space
- Car Park
- Clearance space

The variables that will not be used:

- Store ID
- Manager Name
- Town
- Country

Pre-processing

I entered the data file into weka and using the pre-process tab looked through each of the variables and charts for any errors such as missing data but also looking at the spread of data in the charts to find any outliers, minority values and minority values. I also looked at each variable chart to see if the data was useful as in it had a lot of similar data sets and not a lot of minority values with a flat and wide distribution (1 or 2 of each possible value) as this wouldn't be very useful as there wouldn't be much of a pattern if all the values are different and data mining task is to find patterns from specific data.

Errors Found

Country: looking through the data for countries I found that two entries have been set as France, further research looking into the location of these data set shows that they are location based in the UK so we can assume these are errors in the data and corrected them to UK, this shouldn't affect the overall outcome.

Staff: looking through the staff data I found that multiple values didn't fit with the rest of the data: - 2, 300, 600 these values will be labelled as outliers and will be removed as they are mostly likely errors and if kept can disrupt the data mining process and give misleading results.

Car Park: looking through the carpark there are a few attributes labelled as "Y" /" N" instead of the majority "Yes" or "No" these instances have been changes to fit the usual format of the rest of the data to reduce possible errors / misleading results.

Location: looking through the location data there is only 1 instance of "Village" this isn't very useful as we would need more data for villages for it to be useful in our data predictions. So it will be removed to reduce misleading results.

Changes in the data set:

Data Set 97, 41 Country changed to "UK"

Data Set 4,55,109 removed due to outliners values being entered for number of staff

Data Set 9, 12, 31, 63, 85, 117, 123 have been updated to either "Yes" instead of "Y" or "No" instead or "N"

Data Set 36 removed to only having 1 instance of village

Chart Before

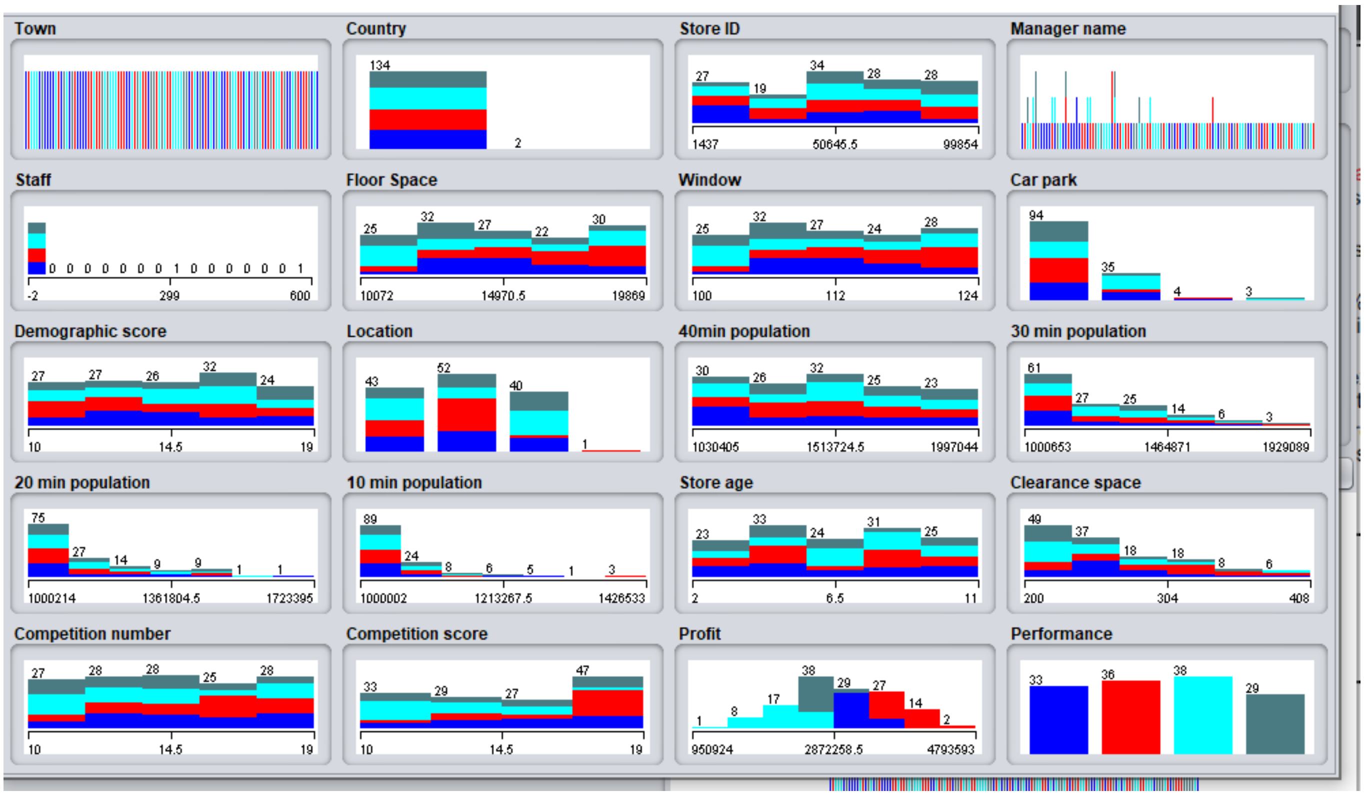
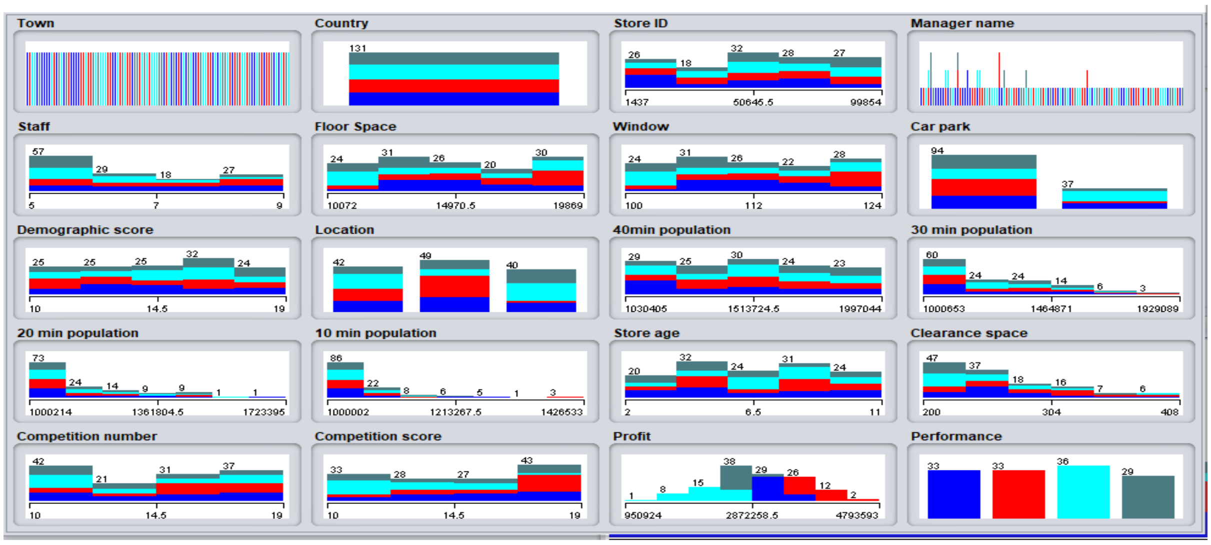


Chart After



Using weka I selected the attributes by clicking the tick box in the pre-process tab and selecting remove, variables removed: Country, Town, Store ID, Manager Name.

Data Mining Algorithms

Decision Tree

A decision tree is a structure that includes a root node, branches and leaf nodes. The root node is the starting point of the tree, the leaf nodes holds a class label/variable like age or credit rating and the branches represent the outcome of a test using the nodes corresponding values that the node takes in and keep branching down until your object is classified. the decision tree works best with nominal attributes that can be classified; numeric values will need to be split into bins.

ID3 Algorithm

The ID3 algorithm splits the nodes based on the greatest information gained. Information is measured based on the probability of something happening. To calculate the information associated with a single event: I(e) = -log(pe) where pe is the probability of event e occurring and log is the base 2 log.

ID3 picks the top node of the tree by calculating the information gain of the output class for each input variable and selects the one that removes the most uncertainty and then creates a branch for each value that variable can take.

To calculate the information gained we first need to calculate entropy, entropy is the weighted average of information across the possible values of a variable. Entropy is calculated by getting the sum of probability of each of the possible events and multiply that information value.

$$H(X) = \sum P(x_i) I(x_i) = -\sum P(x_i) \log(P(x_i))$$

(entropy formula)

After calculating the entropy, we can now calculate the information gain. To calculate information gain we use the decrease of entropy after the data split on a variable, this is known as the condition entropy formula H(Outcome | Known) where the outcome is the predicated output and the known is the predictor that is used. If we know that H(Outcome) and H(Outcome | input) we can calculate how much the input tells up about the outcome using the formal H(Outcome) – H(Outcome | input)

Branches are created by repeating the same information gained calculation based on the location on the tree of the current branch. If all objects at the current leaf are of the same class, then no more branching is needed, and the algorithm stops when all the data has been accounted for.

a decision tree algorithm C4.5 (called j48 in weka) which is a classification algorithm and an extension of the ID3 algorithm is used to generate the decision tree in weka. This algorithm works effectively with noisy data from datasets using a pruning method which removes noise by creating a leaf where the node considered to be noise is located which reduces the classification error of the noisy data.

Pros and cons

Pros	Cons
Easy to visualize if number of levels of the tree are low	Can look messy/ hard to visualize of number of levels are high
Easy to implement	Doesn't work well with numeric data (weka won't allow numeric data predictions)

Performs well with enough data	Expensive to train
Works well at predicting classifications	

Multilayer Perceptron’s

Multilayer perceptron is a type of neural network which can learn a function between our inputs and the outcome. It works by building the function between the inputs which builds a network out of many small simple functions that are joined by weighted connections.

The MLP structure consists of 3 layers input, hidden and output. the input layer which takes in input values, each input value creates an input node. the hidden layer at the start contains a random value but then starts to use the data to modify the weighted connections values which is repeated until it can predict the data accurately, this process is referred as “training the neural network”. The output layer is the difference between the output and the value in the training data which is known as the “error” which is used to find out how accurate our prediction is.

The neural network training consists of 7 steps:

Step 1: Prepare the data, this step takes the data and adds the predictors and the predicted variables and makes sure that each row has an example of each.

Step 2: Split the data into a test set and a training set

Step 3: Read Each row one by one into the neural network showing the predictors as input values and the predicted value as the target output

Step 4: Make a prediction, this step compares the predicted value given by the neural network to the target value

Step 5: Update the weights using backpropagation which balances the weights to reduce the output error

Step 6: Present the next example in the file

Step 7: Repeat until the error can no longer reduce any lower and ideally stop when the test error is at its lowest

The predictors are the variables used to make the prediction and the prediction value is the output value.

Backpropagation

Each neuron is composed of two units 1 is the weighted sum of the input signals, 2. The realization of neuron activation function which represents the output.

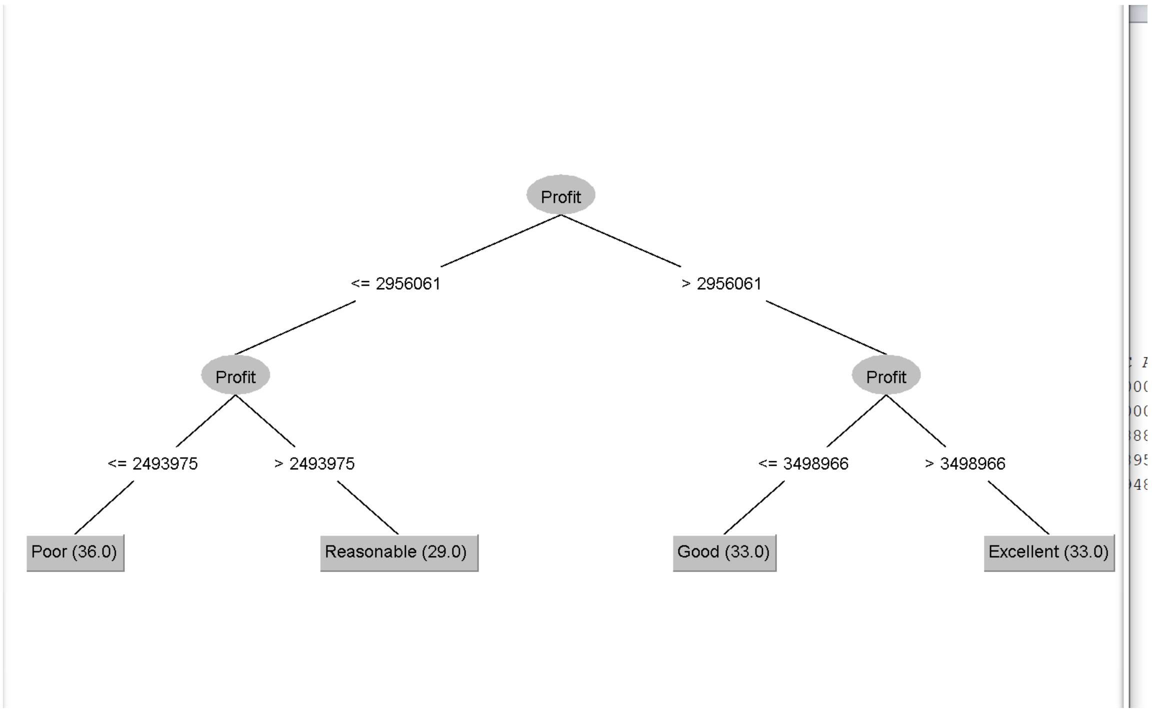
1. Is calculated as the weights of connection between network input x(m) and neuron n input layer.
2. Is calculated as W(xm)n and Y(n)

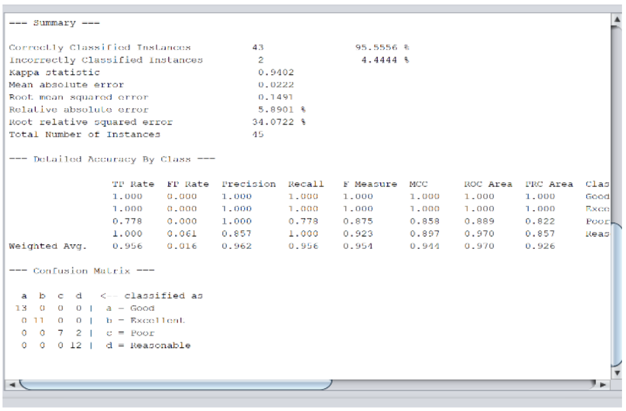
Pros	Cons
Works well with predictions	No rules to look at
Can use numeric and nominal values as inputs	Can make errors if not trained properly
Can generalise data that has not been seen before	

Solution

J48 Algorithm

J48 algorithm isn't useful in this data set as we are predicating profit a numeric value which weka won't allow you to run, we can switch to performance but this too isn't useful as J48 will use the profit as it's root node to predict performance which isn't what we want since you shouldn't use the value you want predicated as an input value, since J48 is for classification or in other words prediction of the present I can't recommend using J48 Algorithm in this problem since it won't be able to accurately predict profit with the data set I was given.





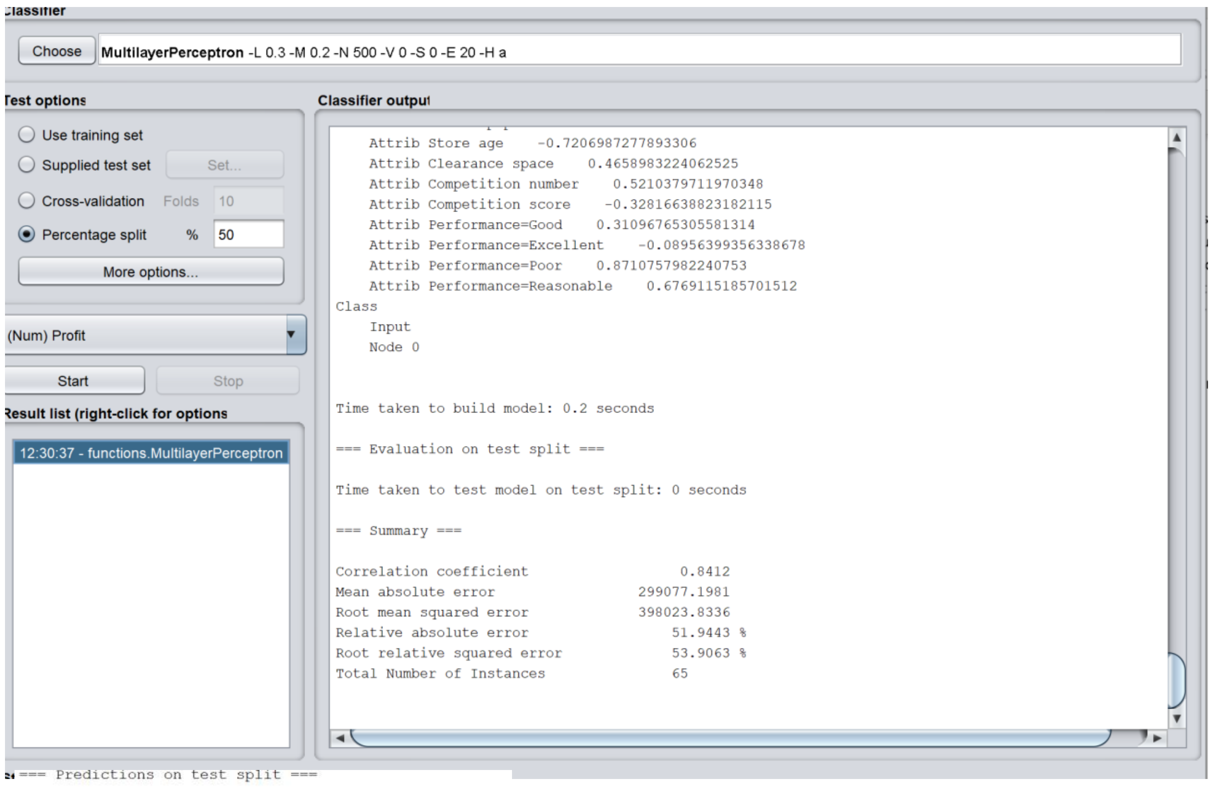
```
=== Summary ===

Correctly Classified Instances          43               95.5556 %
Incorrectly Classified Instances         2                4.4444 %
Kappa statistic                          0.9402
Mean absolute error                      0.0222
Root mean squared error                  0.1491
Relative absolute error                  5.8901 %
Root relative squared error             34.0722 %
Total Number of Instances               45

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F Measure  MCC      ROC Area  PRC Area  Clas
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Good
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Exce
                 0.778    0.000    1.000      0.778    0.875      0.858    0.889     0.822     Poor
                 1.000    0.061    0.857      1.000    0.923      0.897    0.970     0.857     Reas
Weighted Avg.    0.956    0.016    0.962      0.956    0.954      0.944    0.970     0.926

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 13  0  0  0 |  a = Good
  0 11  0  0 |  b = Excellent
  0  0  7  2 |  c = Poor
  0  0  0 12 |  d = Reasonable
```

Looking at the confusion matrix we can see that it got a 95% accuracy rating which is a good result but made errors when identifying 2 of the poor performance predictions ("c).

Multilayer Perceptron

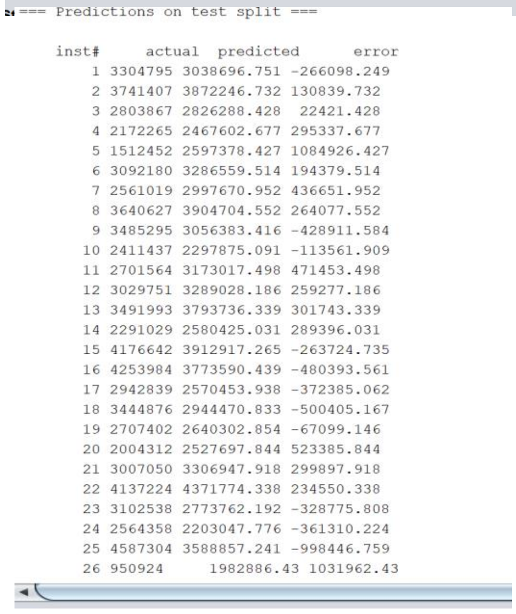
I investigated if a 50 % split, cross validation or training set would give the best prediction accuracy with the lowest mean absolute error aiming to get below the 500k mark using multilayer perceptron.

The 50-percentage split splits the data evenly, 1 side being used to process the data and the other used for testing. I used the default settings and found that the 50 % split gave the best possible result with just under a 300k mean absolute error result



```
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
```

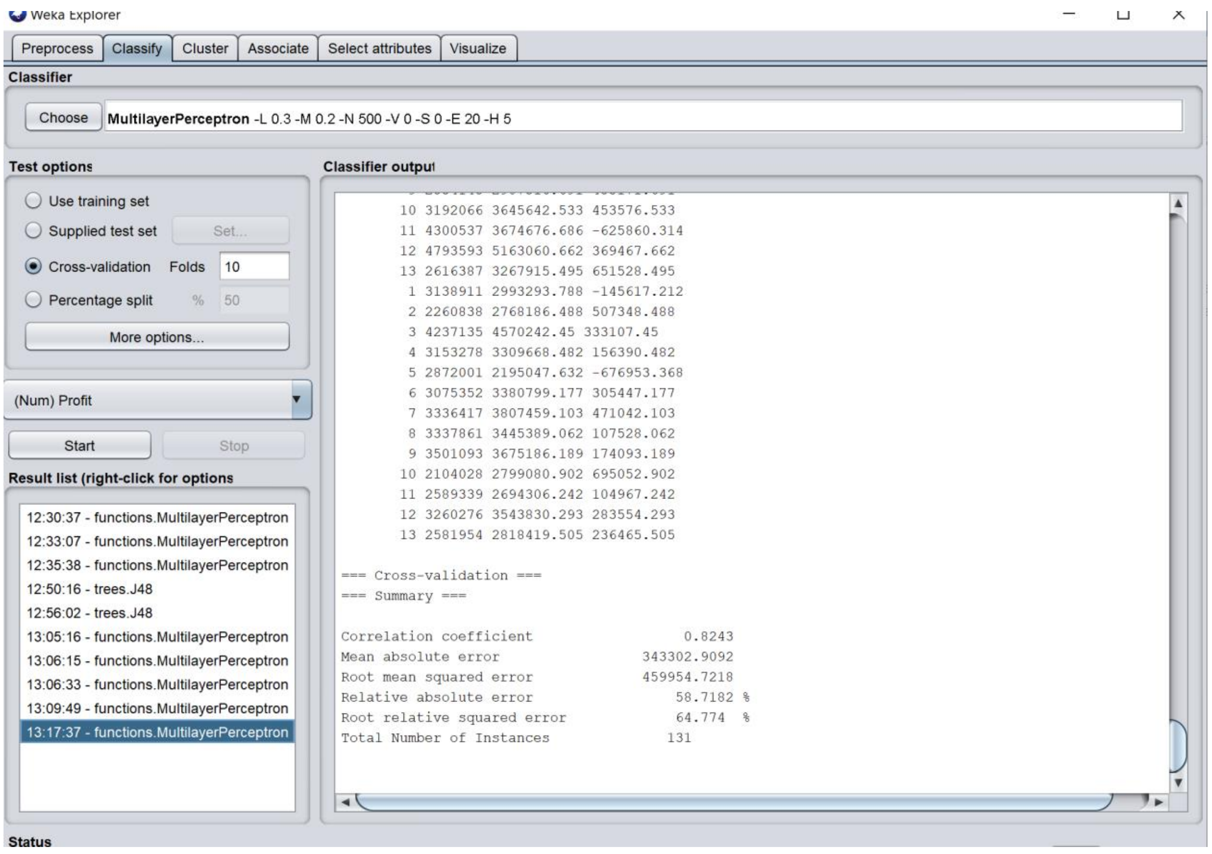
```
    Attrib Store age    -0.7206987277893306
    Attrib Clearance space    0.46589832240625 25
    Attrib Competition number    0.5210379711970348
    Attrib Competition score    -0.32816638823182115
    Attrib Performance=Good    0.31096765305581314
    Attrib Performance=Excellent    -0.08956399356338678
    Attrib Performance=Poor    0.8710757982240753
    Attrib Performance=Reasonable    0.6769115185701512
Class
    Input
    Node 0


Time taken to build model: 0.2 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.8412
Mean absolute error                 299077.1981
Root mean squared error             398023.8336
Relative absolute error                 51.9443 %
Root relative squared error             53.9063 %
Total Number of Instances               65
```



```
=== Predictions on test split ===

    inst#     actual  predicted      error
        1 3304795 3038696.751 -266098.249
        2 3741407 3872246.732 130839.732
        3 2803867 2826288.428  22421.428
        4 2172265 2467602.677 295337.677
        5 1512452 2597378.427 1084926.427
        6 3092180 3286559.514 194379.514
        7 2561019 2997670.952 436651.952
        8 3640627 3904704.552 264077.552
        9 3485295 3056383.416 -428911.584
       10 2411437 2297875.091 -113561.909
       11 2701564 3173017.498 471453.498
       12 3029751 3289028.186 259277.186
       13 3491993 3793736.339 301743.339
       14 2291029 2580425.031 289396.031
       15 4176642 3912917.265 -263724.735
       16 4253984 3773590.439 -480393.561
       17 2942839 2570453.938 -372385.062
       18 3444876 2944470.833 -500405.167
       19 2707402 2640302.854 -67099.146
       20 2004312 2527697.844 523385.844
       21 3007050 3306947.918 299897.918
       22 4137224 4371774.338 234550.338
       23 3102538 2773762.192 -328775.808
       24 2564358 2203047.776 -361310.224
       25 4587304 3588857.241 -998446.759
       26 950924      1982886.43 1031962.43
```

Cross Validation

this method folds the data in 10 folds and uses 9 of the folds for learning and leaves 1 of the folds for testing this process is repeated for however many folds leaving a different fold out for testing.



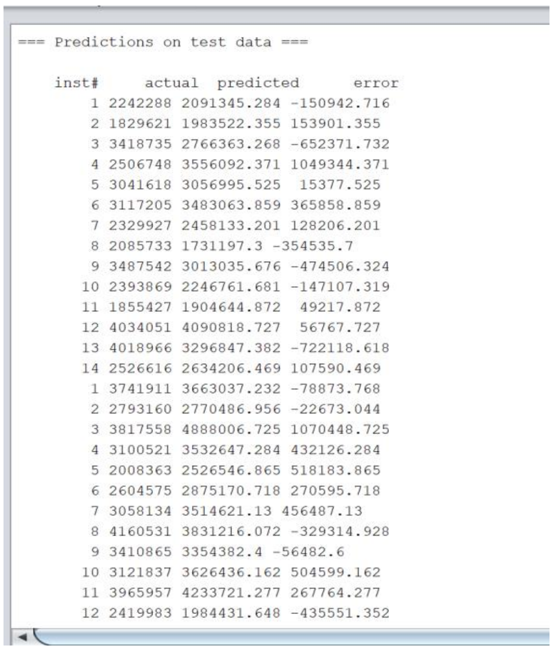
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H 5

Use training set
Supplied test set
[✓ Cross-validation] Folds 10
Percentage split % 50

(Num) Profit

12:30:37 - functions.MultilayerPerceptron
12:33:07 - functions.MultilayerPerceptron
12:35:38 - functions.MultilayerPerceptron
12:50:16 - trees.J48
12:56:02 - trees.J48
13:05:16 - functions.MultilayerPerceptron
13:06:15 - functions.MultilayerPerceptron
13:06:33 - functions.MultilayerPerceptron
13:09:49 - functions.MultilayerPerceptron
13:17:37 - functions.MultilayerPerceptron

```
       10 3192066 3645642.533 453576.533
       11 4300537 3674676.686 -625860.314
       12 4793593 5163060.662 369467.662
       13 2616387 3267915.495 651528.495
        1 3138911 2993293.788 -145617.212
        2 2260838 2768186.488 507348.488
        3 4237135 4570242.45 333107.45
        4 3153278 3309668.482 156390.482
        5 2872001 2195047.632 -676953.368
        6 3075352 3380799.177 305447.177
        7 3336417 3807459.103 471042.103
        8 3337861 3445389.062 107528.062
        9 3501093 3675186.189 174093.189
       10 2104028 2799080.902 695052.902
       11 2589339 2694306.242 104967.242
       12 3260276 3543830.293 283554.293
       13 2581954 2818419.505 236465.505

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.8243
Mean absolute error                 343302.9092
Root mean squared error             459954.7218
Relative absolute error                 58.7182 %
Root relative squared error             64.774  %
Total Number of Instances              131
```



```
=== Predictions on test data ===

    inst#     actual  predicted      error
        1 2242288 2091345.284 -150942.716
        2 1829621 1983522.355 153901.355
        3 3418735 2766363.268 -652371.732
        4 2506748 3556092.371 1049344.371
        5 3041618 3056995.525  15377.525
        6 3117205 3483063.859 365858.859
        7 2329927 2458133.201 128206.201
        8 2085733 1731197.3 -354535.7
        9 3487542 3013035.676 -474506.324
       10 2393869 2246761.681 -147107.319
       11 1855427 1904644.872  49217.872
       12 4034051 4090818.727  56767.727
       13 4018966 3296847.382 -722118.618
       14 2526616 2634206.469 107590.469
        1 3741911 3663037.232 -78873.768
        2 2793160 2770486.956 -22673.044
        3 3817558 4888006.725 1070448.725
        4 3100521 3532647.284 432126.284
        5 2008363 2526546.865 518183.865
        6 2604575 2875170.718 270595.718
        7 3058134 3514621.13 456487.13
        8 4160531 3831216.072 -329314.928
        9 3410865 3354382.4 -56482.6
       10 3121837 3626436.162 504599.162
       11 3965957 4233721.277 267764.277
       12 2419983 1984431.648 -435551.352
```

Results

I ran the MLP test on the profit prediction and J48 on the performance classification

Table of results:

Model	50:50 Split	70:30 Split	Cross Validation
MLP	301309.087	499619.7056	360168.2406
J48	96.9231%	94.8718%	97.7099%

From these results the data for the j48 shows a very high accuracy rating but this is for classification of the performance prediction so it's not very realistic for prediction the profit. Running the MLP Tests showed good results as well as the mean error was below the target 500k mark on each of these tests and I found that the best over all results came from the 50% split.

Recommendations

Variable Recommendations

You should not use Store ID, Manager Name, Country or Town for reasons mentioned above

For testing the variables, I removed each variable to see if it made if it increased or decreased my error rating of 29907.1981

Location was tested first as it was stated to be the most expensive data set to collect

Variable Removed	Mean Error
40 min Location	321729.9086
30 min Location	263376.7813
20 min Location	336179.5903
10 min Location	322256.043
40,30,20,10 min location	370360.548
20,10 min location	316003.1829
40, 30 min location	305732.0843
30,10 min location	337948.9383
20, 40 min location	303835.2898
10, 40 min location	294539.8391
20,30 min location	356078.9284

Other variables tested

Variable Removed	Mean Error
Staff	274536.7078
Floor Space	320146.7536
Window	314962.9273
Car Park	378121.6913
Demographic Score	325061.748
Location	318612.8301
Store Age	351923.3331
Clearance Space	269829.7662
Competition Number	308401.9519
Competition Score	390197.387

Results:

Testing the location data, I found that the biggest difference came from removing the 30 min location which reduced my error rating to 263376.7813, the other variables increased my error rating when removed apart from removing 10, 40 min location which also reduced my error rating to 294539.8391 so judging from that I would recommend removing these two variables and keeping 30/20 min since the difference is minimal and saving the cost of two sets of data would be more beneficial, you could also remove all the location data which increase the error but not by a large margin and still under the 500k mark but if you want the most accuracy I recommend removing the two variables 10/40 min.

Looking at the other variables I found that removing staff and clearance space reduced my error rating so I ran a test removing the 10/40 min location staff and clearance space which gave me an error rating of 297443.6209 with a 50% split test, this is still a good result as it reduced my error rating from my original test and removed 4 variables in the process reducing the cost of data collection so I would recommend removing the variables:

- 10 min Location
- 40 min Location
- Staff
- Clearance Space.

This method isn't ideal for testing if you were working with a very large data set with a few hundred variables it would take you a very long time to shift through all that data but since this is a small data set I was able to do this test in a minimal amount of time but would not recommend if working with a large data set.

Recommendation of Algorithm to use

After testing both the J48 and MLP algorithms I would recommend using the MLP algorithm as it gave a very accurate predication result with a low error rate. J48 doesn't work for this prediction as it can't predict the future only the present as it is a classification algorithm, so I won't recommend using this algorithm.

Summary of Results

After taking out the variables I recommended be removed I Re-ran the MLP algorithm to get a final set of results using the previous test methods:

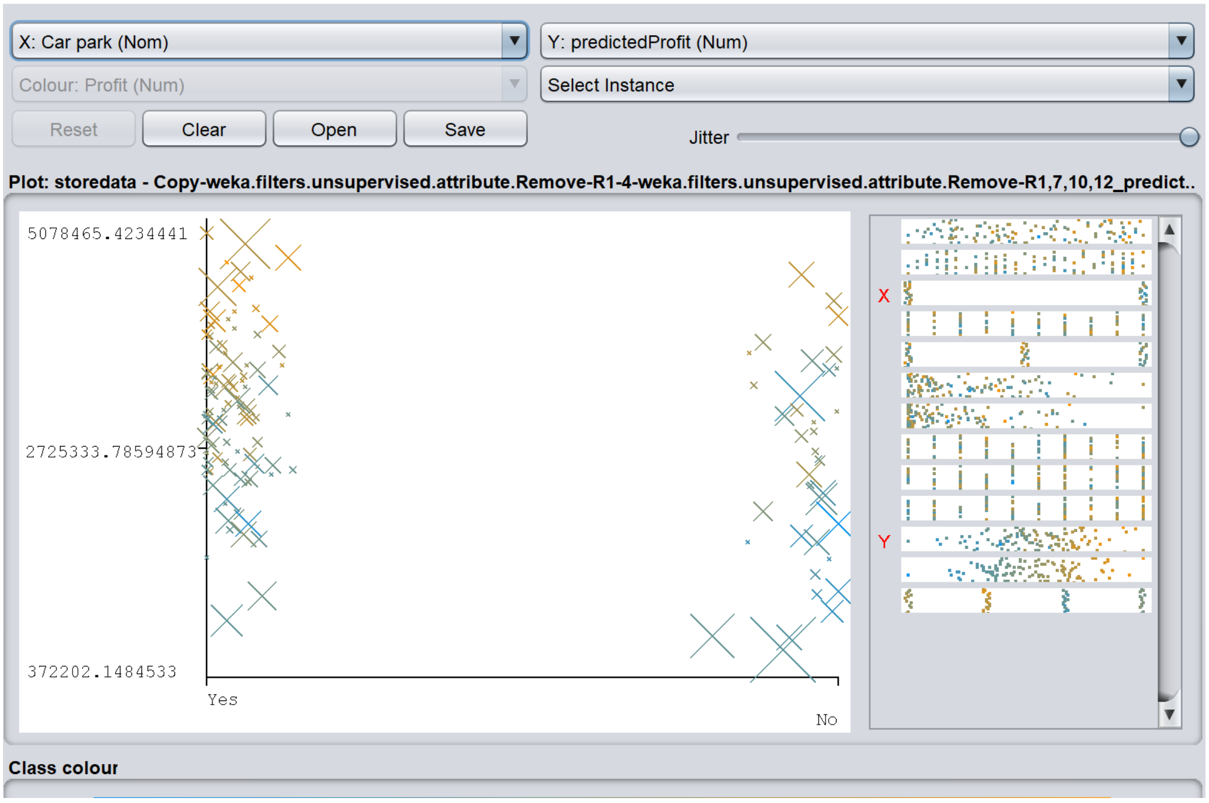
MLP Results	Cross-Validation 10-fold	50:50 Split
Correlation Coefficient	0.8252	0.8293
Root Mean Squared Error	485775.861	412683.741
Mean Absolute error	360168.2406	29744.6209
Relative Absolute error	61.6028%	52.332%
Root Relative squared Error	68.4075%	55.8918

From running these new tests, I found that the 50% split was still more accurate than the 10-fold testing method.

Data Analysis

To find out why some stores are doing better than others I looked at the visualized charts to see what variables data makes the most difference to the profit using that we can see why other stores perform better than others

Car Park:


X: Car park (Nom)

Y: predictedProfit (Num)

Colour: Profit (Num)

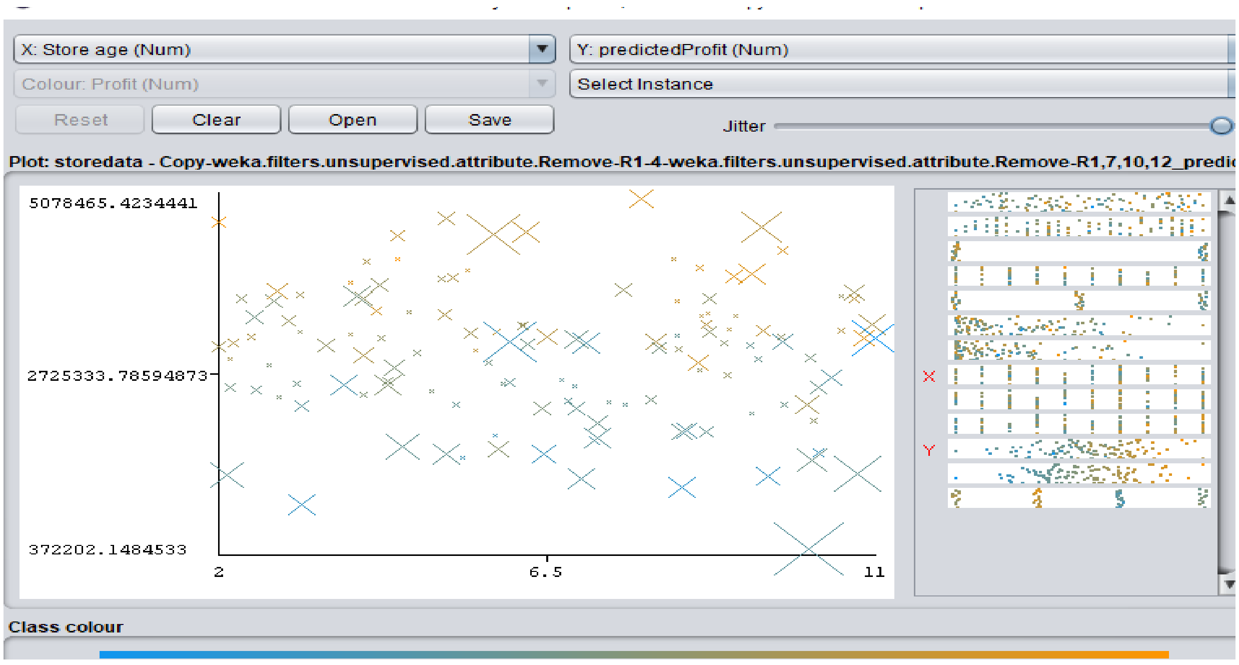
Select Instance

Plot: storedata - Copy-weka.filters.unsupervised.attribute.Remove-R1-4-weka.filters.unsupervised.attribute.Remove-R1,7,10,12_predict..

Class colour

From the data we can see that stores with car parks performed but better than stores without showing that having a car park is a big contributing factor for profits.

Store Age:


X: Store age (Num)

Y: predictedProfit (Num)

Colour: Profit (Num)

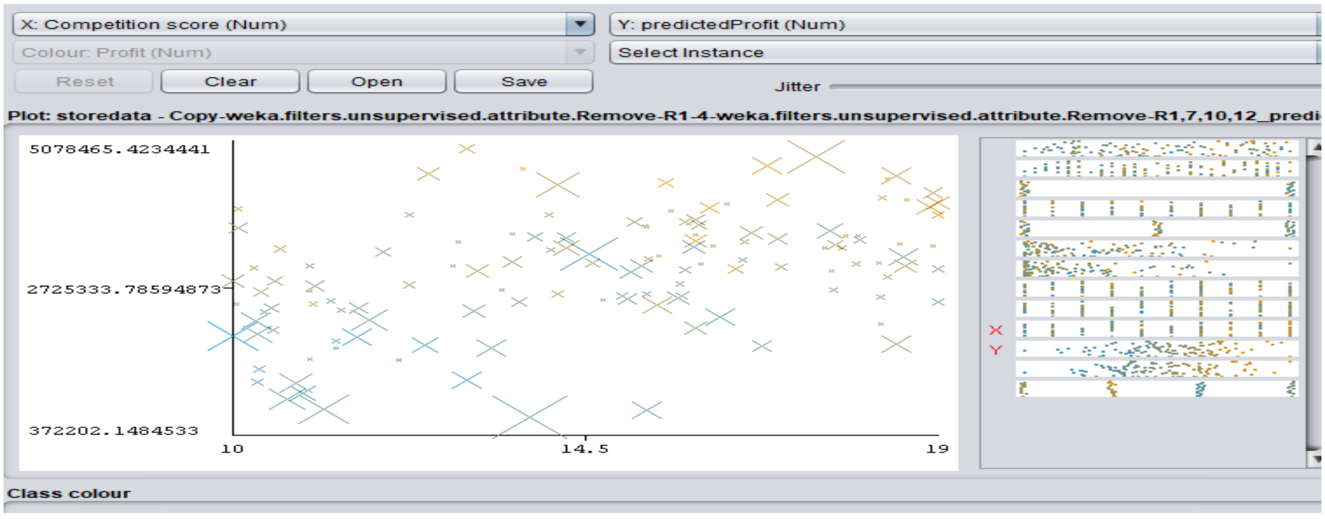
Select Instance

Plot: storedata - Copy-weka.filters.unsupervised.attribute.Remove-R1-4-weka.filters.unsupervised.attribute.Remove-R1,7,10,12_predi

Class colour

From the data we can see that typically the majority of older stores don't make a large amount of profit similarly to stores that have only been open for 2 years with the majority of profits going to stores between 4-7 years.

Competition Score:


X: Competition score (Num)

Y: predictedProfit (Num)

Colour: Profit (Num)

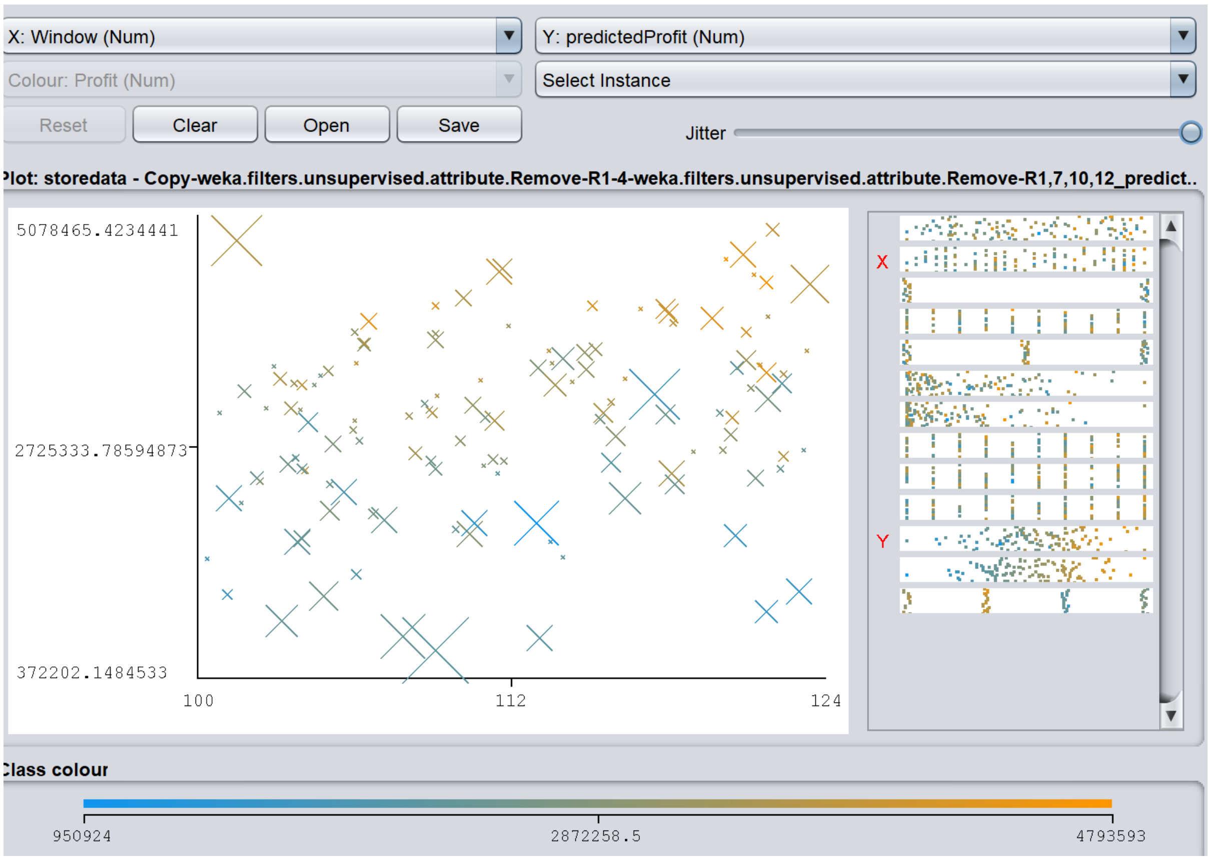
Select Instance

Plot: storedata - Copy-weka.filters.unsupervised.attribute.Remove-R1-4-weka.filters.unsupervised.attribute.Remove-R1,7,10,12_predi

Class colour

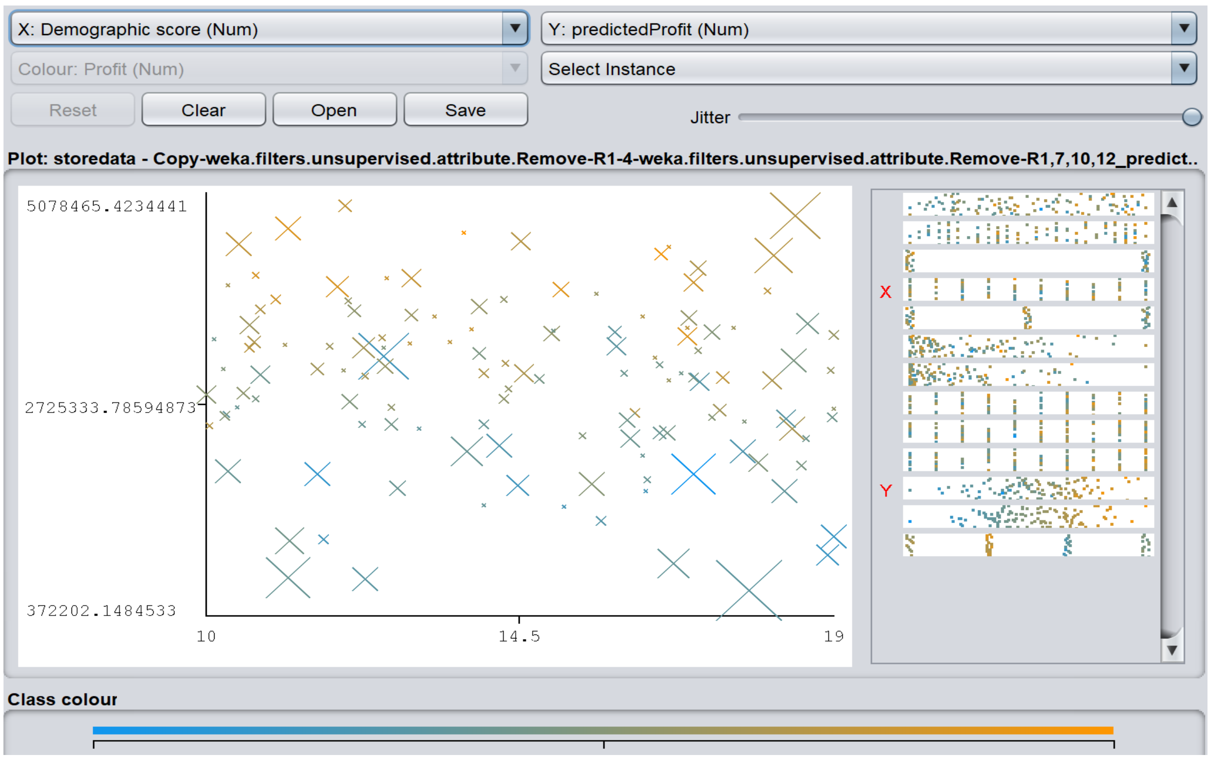
From the data we can see the lower the competition score doesn't increase our stores profit as you might of thought but instead, we see that the lower our competition score is the profit scores are lower as well and increases as our competition score does as well

Window :



From the window data we can see a correlation between the size of the window and the profit growth so we can see the bigger the windows the bigger the profit in most cases.

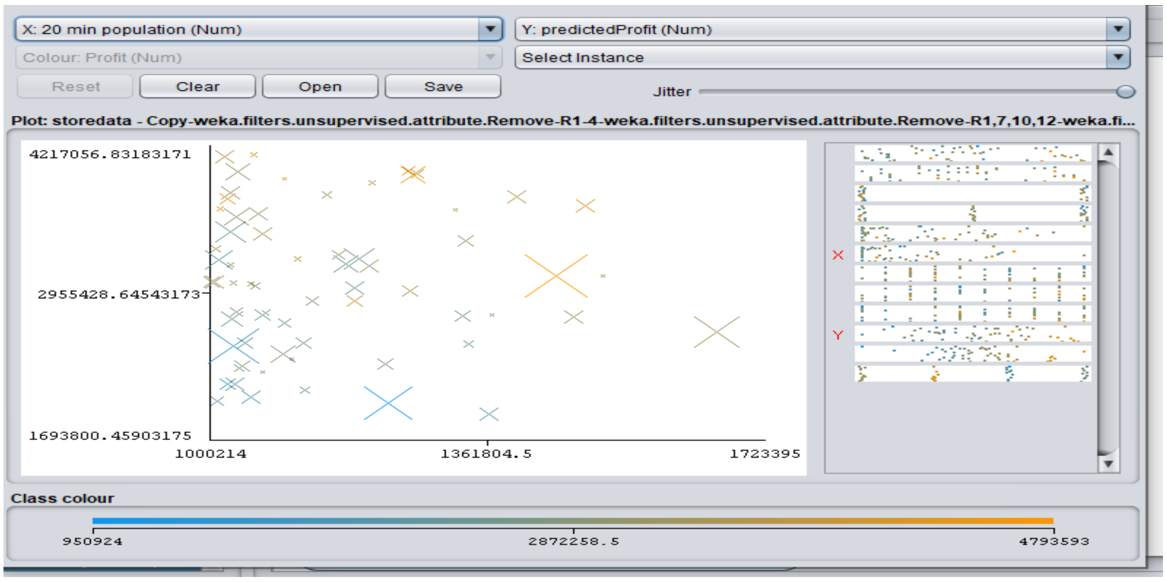
Demographic Score



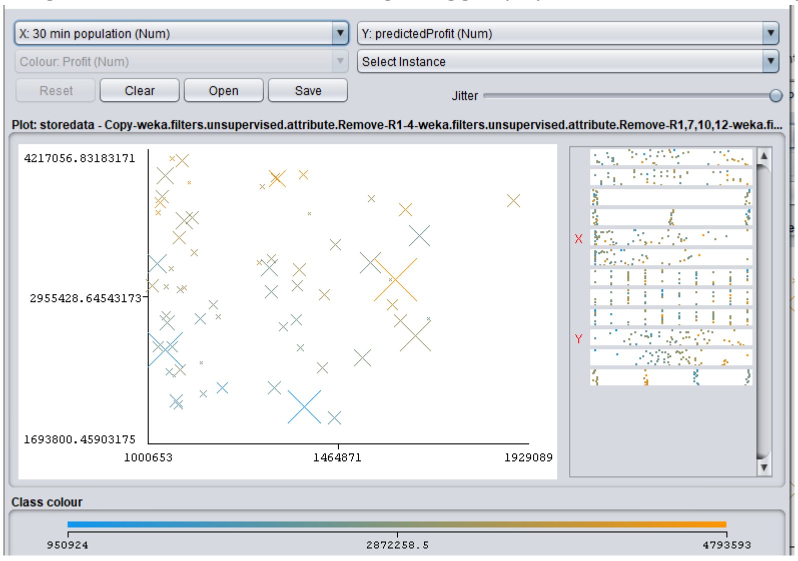
Plot: storedata - Copy-weka.filters.unsupervised.attribute.Remove-R1-4-weka.filters.unsupervised.attribute.Remove-R1,7,10,12_predict..

From the demographic score data we can see there is little to no correlation between the score and the profit growth noting that the profit doesn't necessary increase with higher demographic scores so it would be a good idea not to include this data

Rerunning the 50% split increased the mean absolute error to 339406.7692 which might increase our error rating but gives us a more accurate result

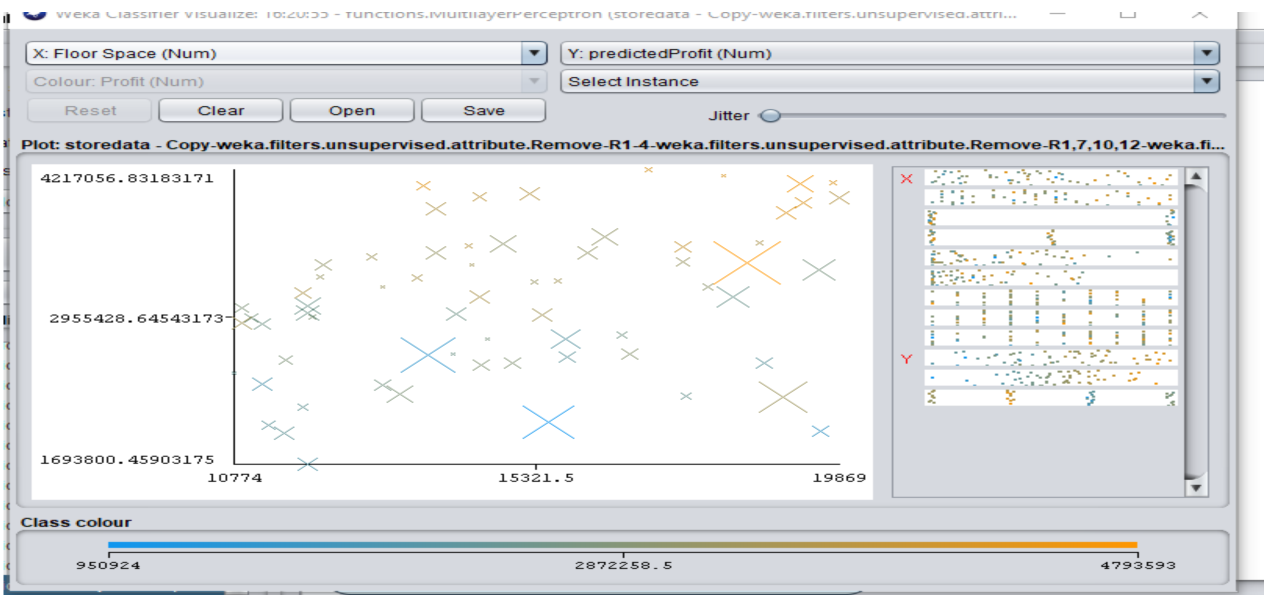


Looking at the 20 min population data most of the profit's stores come from within the population range of a million and not having a bigger population means having a larger profit.



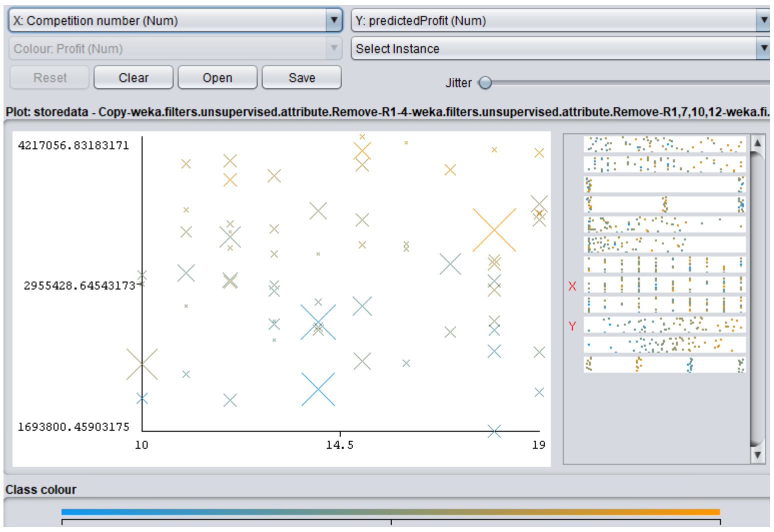
Similarly, to the 20 min range we see that most of the profit comes from the 1 million range and there is no correlation between having a larger population size means having a larger profit score

Floor Space:


Weka Classifier Visualize: 16:20:55 - functions.MultilayerPerceptron (storedata - Copy-weka.filters.unsupervised.attri...
X: Floor Space (Num)
Y: predictedProfit (Num)
Colour: Profit (Num)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: storedata - Copy-weka.filters.unsupervised.attribute.Remove-R1-4-weka.filters.unsupervised.attribute.Remove-R1,7,10,12-weka.fi...
4217056.83183171
2955428.64543173
1693800.45903175
10774
15321.5
19869
Class colour
950924
2872258.5
4793593

From this data we can see a direct correlation between the floor space available and the profits, noting that no store with little floor space of 10774 did excellent but a large portion of stores with a large floor space did perform excellent.

Competition num


X: Competition number (Num)
Y: predictedProfit (Num)
Colour: Profit (Num)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: storedata - Copy-weka.filters.unsupervised.attribute.Remove-R1-4-weka.filters.unsupervised.attribute.Remove-R1,7,10,12-weka.fi.
4217056.83183171
2955428.64543173
1693800.45903175
10
14.5
19
Class colour

Looking at the competition we can see that stores with a low number of competitor stores nearby do poorly while increases when the number of competition stores increases similar to the compition score we can see that profit isn't increased if there are competitors near by but the opposite is true

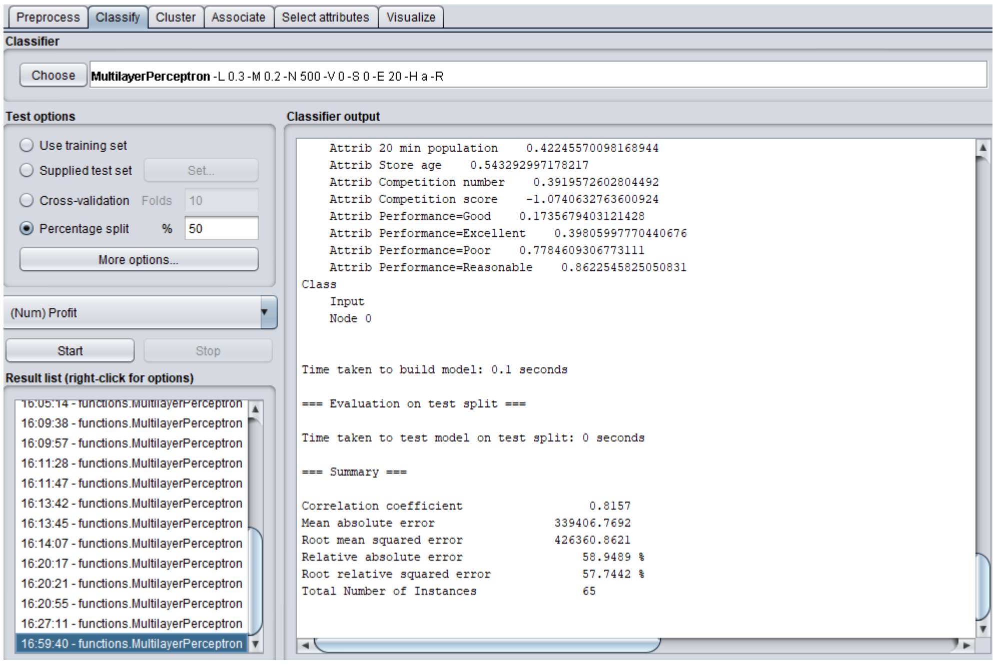
Profit and performance do not need to be viewed as they hold no meaningful analysis for identifying correlations for profits predictions.

From the charts we where able to see a direct correlation for most of the variables and we can now estimate why other stores are preforming better than others.

Changes

Demographic variable removed.

Results after changes


Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
50
More options...
(Num) Profit
Start
Stop
Result list (right-click for options)
16:05:14 - functions.MultilayerPerceptron
16:09:38 - functions.MultilayerPerceptron
16:09:57 - functions.MultilayerPerceptron
16:11:28 - functions.MultilayerPerceptron
16:11:47 - functions.MultilayerPerceptron
16:13:42 - functions.MultilayerPerceptron
16:13:45 - functions.MultilayerPerceptron
16:14:07 - functions.MultilayerPerceptron
16:20:17 - functions.MultilayerPerceptron
16:20:21 - functions.MultilayerPerceptron
16:20:55 - functions.MultilayerPerceptron
16:27:11 - functions.MultilayerPerceptron
16:59:40 - functions.MultilayerPerceptron
Classifier output

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -R

Percentage split [✓] % 50

```
    Attrib 20 min population    0.42245570098168944
    Attrib Store age    0.543292997178217
    Attrib Competition number    0.3919572602804492
    Attrib Competition score    -1.0740632763600924
    Attrib Performance=Good    0.1735679403121428
    Attrib Performance=Excellent    0.39805997770440676
    Attrib Performance=Poor    0.7784609306773111
    Attrib Performance=Reasonable    0.8622545825050831
Class
    Input
    Node 0


Time taken to build model: 0.1 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.8157
Mean absolute error                 339406.7692
Root mean squared error             426360.8621
Relative absolute error                 58.9489 %
Root relative squared error             57.7442 %
Total Number of Instances               65
```