Data Mining Assignment

CSCU9T6 – SPRING 2020

2636515
UNIVERSITY OF STIRLING

Introduction

The CEO of a chain of shops spread throughout the UK has approached me and requested the delivery of two models which aim to assist in the effective running of these stores. The shops themselves are similar yet at the same time exhibit varying degrees of revenue success with figures of between £1 and £5 million per annum. Each of the models will be used to fulfil specific requirements that have been set by the company. The first is to predict and estimate the revenue that a store should generate with the second model being implemented to classify any existing stores into distinct categories of performance.

- Estimating the predicted revenue will be used to evaluate the entire chain of stores helping to seek out locations that may be underperforming. With this knowledge corrective action can be taken to address the issues found through the data mining model.
- The other model will aid in clearly classifying all existing stores into performance categories: Poor, Reasonable, Good, Excellent. This will further improve the efficiency of running the company and help management better understand why some stores perform better than others.

Data mining is the process used to extract usable data from a larger set of raw data. It can find patterns and correlations amongst the many attributes in a data set. From this definition, we can conclude that using data mining processes is a perfect suit in equipping the company with the tools to help them better run the chain of stores.

Following consultation with the company and with the help of appropriate data mining techniques this report sets out to do the following:

- Investigate the dataset provided by the company
- Describe the data preparation steps taken to clean the raw data
- Detail both the multi-layer perceptron and decision tree techniques to be used in the modelling process
- Construct the models and evaluate their results
- Make recommendations based on the stronger performing modelling technique and the set of variables that should and shouldn't be collected in future

Data Summary

World of Bargains has provided data for each of its stores in a CSV file containing 136 shop instances each defined through 20 separate attributes.

Attributes:

- *nom = nominal num = numeric*

Inputs = *Town (nom), Country (nom), Store ID (num), Manager Name (nom), Staff (num), Floor Space (num), Window Space (num), Car Park (nom), Demographic Score (num), Location (nom), 40min Population (num), 30min Population (num), 20min Population (num), 10min Population (num), Store Age (num) , Clearance Space (num), Competition Number (num), Competition Score (num)*

Outputs = *Profit (num), Performance (nom)*

Distribution features that must be accounted for:

- **Outliers**
  - It is an observation point that is distant from other observations
  - Being the most extreme observations, they may include the sample maximum or minimum
  - They can disrupt the data mining process and give misleading results
  - In most cases, they should be removed from the data set
- **Minority Values**
  - Values that only appear infrequently throughout the data
  - If they don't appear often enough to contribute to the model, they should be removed from the data
- **Data Entry Errors**
  - Outliers and Minority Values could also be the result of Data Entry Errors
  - Inputting the incorrect data is amongst the most typical data problem
  - An unintentional error as mentioned above can also disrupt the data mining process and should be removed
- **Flat and Wide Variables**
  - Attributes in which all the values have a flat, wide distribution
  - These variables are of little use in data mining as they fail to find general patterns from their respective data and should be excluded from the model

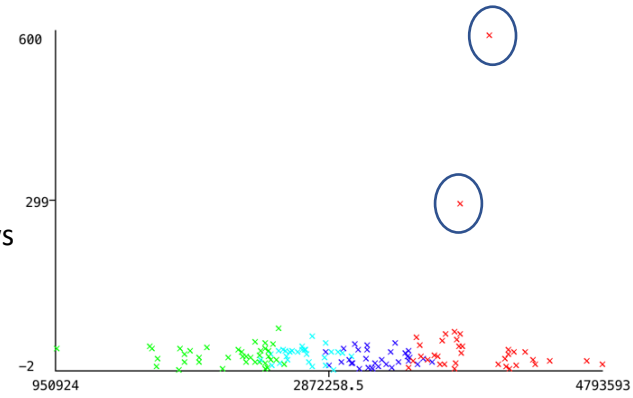
Data Preparation

When attempting data mining many parameters can be adjusted to optimise the running of the algorithm used to model the data. However, if the data given is not appropriate to the task the modelling phase will be unsuccessful in establishing the best possible solution. Therefore, the raw data must undergo preparation to be deemed sufficiently good enough to carry out the task of predicting revenue and classifying store performance. Instead of manually preparing the data through Excel, for example, the data mining software Weka will be utilised to better and faster implement these steps internally. This removes much of the pre-processing minimising most of the manual adjustments needed to improve the model. Any manual data manipulation steps taken will be detailed below.

Removing rows and adjusting incorrect data

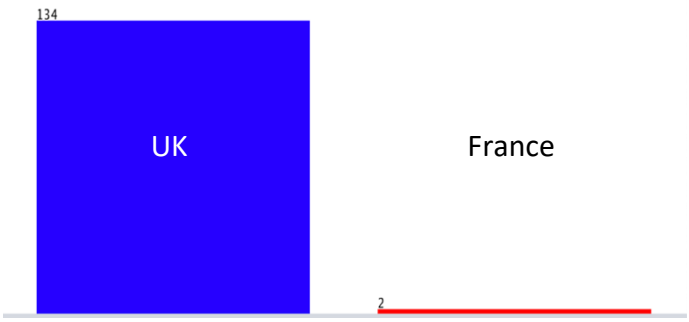
Staff:

The Staff attribute as can be seen from the graph has two outlier data points (Staff numbers of 600 and 300) that are very distant from the rest of the dataset. These points would disrupt the data mining process and in doing so misrepresent what the data is trying to say. Removing these points therefore allows for a better portrayal of the staff attribute. Further analysis shows another outlier point at the sample minimum of -2. This is most likely a data entry error since having a negative number of staff members at any given store is impossible. This row is also to be removed from the dataset along with the other two rows.



Country:

Within the data, there are 2 distinct values for the Country attribute: UK and France. The histogram below highlights that the vast majority of the values are within the UK (134) with the remaining 2 store entries being located in France. After closer examination, the towns of the 2 entries were found to be geographically in the UK and not France. This and the fact that in the brief there is no mention of any stores outside the UK mean that the two entries are most likely the result of data entry errors. The two entries instances will be amended from France to the UK.



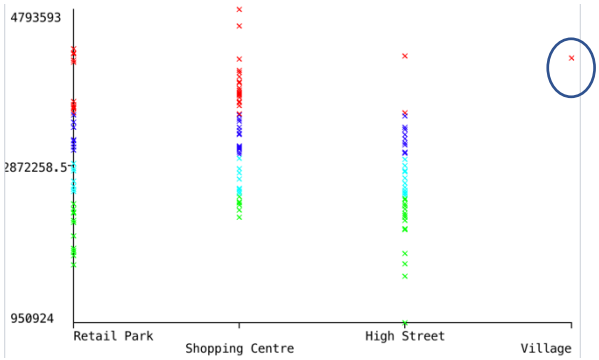
Car Park:

After analysing the dataset for the car park attribute a total of 4 values are used to represent yes and no. The table shows us 'Yes' is the most common value for describing if a store has a car park with 'No' used the most to represent if a store doesn't have a car park. The remaining attribute variations of 'Y' and 'N' are simply to be replaced by 'Yes' and 'No' accordingly.

No.	Label	Count
1	Yes	92
2	No	34
3	Y	4
4	N	3

Location:

As seen in the graph there is a minority value entry in which one store's location is listed as 'Village'. This could disrupt the data modelling and after further investigation the entry is not deemed sufficiently important to be included in the model. Therefore for this store instance, the row will be removed from the dataset.



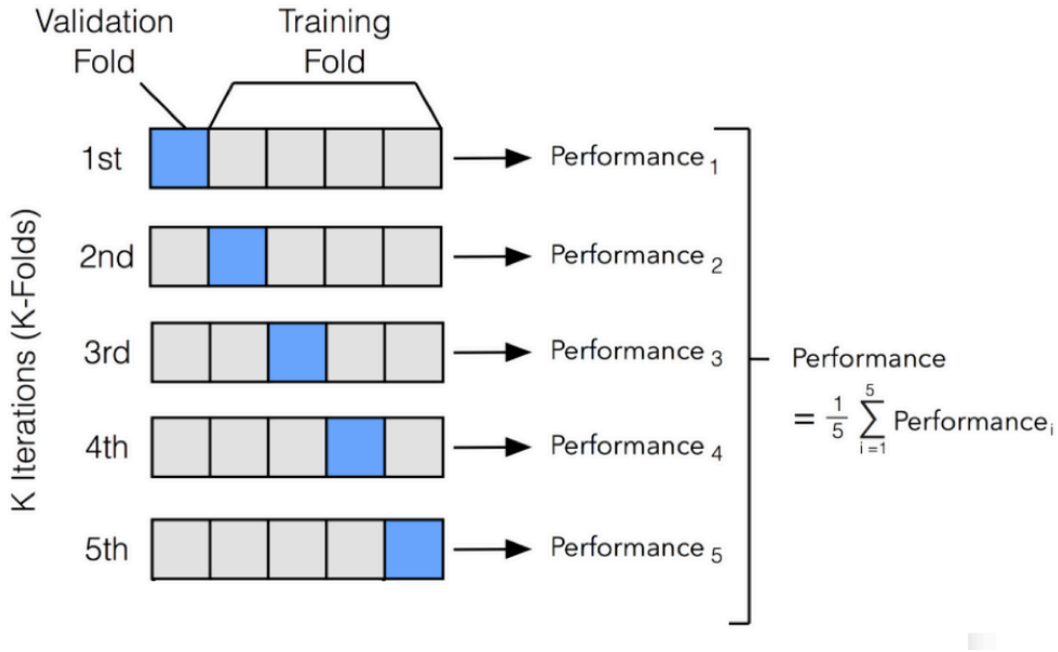
Removing entire attributes from the dataset

During the pre-processing exploration of each variable value in Weka four attributes were found to have no impact on the performance of the model. These variables fall under the distribution of flat or wide variables that was detailed earlier.

- The '*Town*' feature consisted of unique entries 100% of the time so would add no value to the performance of the model
- The '*Country*' feature after being amended is only represented by a single value (UK) throughout all store instances.
- These two features are to be removed from the dataset and in doing so assumes that the model built only predicts and classifies stores within the UK and that no two stores exist within the same town.
- Additionally, the attribute labelled '*Manager Name*' is removed as it serves no purpose being in the model only describing that every store is managed by a single dedicated manager
- Finally, the '*Store ID*' attribute is removed leaving the assumption that every store is assigned a unique ID number and that it plays no part in the performance of the model

Training

To effectively measure each of the model's performance K Cross-Validation will be used. Cross-validation is a resampling procedure used to evaluate models on a limited data sample. This results in a less biased estimate of the model than a simple train/test split making it the perfect choice to be applied on the World of Bargains dataset. As seen in the image below the training dataset provided is subdivided into K subsets. K is commonly chosen as 10 folds in large datasets but with the World of Bargains dataset only consisting of a limited number of instances a smaller K value of 5 is usually considered more appropriate. In each round after splitting the dataset into K parts, one part is used for validation and the remaining K – 1 part are merged into a training subset for the model evaluation. This method results in 5 training sets evaluated on validation sets to calculate an average error. This allows the final model to be evaluated to a high-performing level ensuring the best possible solution is found.



The most common ratio between the training fold and the validation fold is the 70/30 rule where 70% of the entire dataset is used for training and the remaining 30% is set aside for validation. Applying this to the dataset provided, store records 1-92 are to be used in Weka as the training set with the remaining store records 93-132 set aside for the test set.

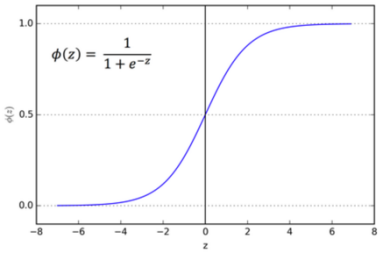
Modelling Techniques

Multi-Layer Perceptron (MLP)

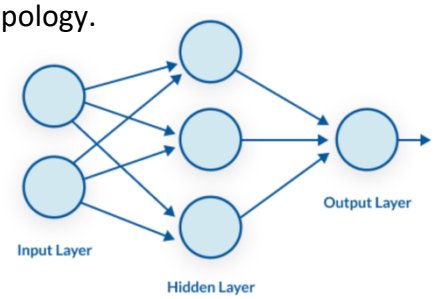
It is an area that explores how simple models of biological brains are used to solve complex tasks like the predictive modelling seen to complete requirements in this report for example. An MLP consists of a network of single neurons called *perceptrons*. Each perceptron computes a single output from multiple real inputs by forming a linear combination according to its input weights. The output is then put through a nonlinear activation function that controls the threshold at which the neuron is activated and the strength of the output signal.

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(\mathbf{w}^T \mathbf{x} + b)$$

where **w** defines the vector of weights, **x** is the vector of inputs, **b** is the bias and **Þ** is the activation function. The bias of each neuron can be thought of as an input and in doing so must also be weighted. Therefore, if a neuron has 4 inputs it requires 5 weights: one for each input and one for the bias. Weights are often represented by smaller values with larger weight values increasing the complexity and fragility of the network. Nowadays in multilayer networks, the most commonly used activation function is chosen to be the *logistic sigmoid* which follows an S curve shape which saturates to 0 or 1 when the input is very small or very large respectively. Other functions like the hyperbolic tangent can be used instead but what they share is their non-linearity which allows the network model more strongly.



However, these single-layer networks are rather limited in the mappings they can perform and so are arranged into a network of perceptrons instead of existing in parallel. Each row of perceptrons is defined as a layer with the entire network having multiple layers that make up the network topology.



Input Layers- takes input from the dataset and in doing so makes it the exposed part of the network (visible layer)
Hidden Layers- takes the inputs which have been weighted and passes them through an activation function of choice to be outputted
Output Layers- responsible for outputting values that fit the format the problem required.

Now the network is ready to be trained on the dataset, but the raw data must first be prepared as outlined earlier in the report. After data preparation a training algorithm is used to train the model and as detailed earlier would be done through *K-fold Cross-Validation.* The weights in the network can be updated to help the learning process of the model and this is controlled by configuring the given parameters: *learning rate* and *momentum.* Once the network has been successfully trained it can be used to make predictions on the test and validation datasets in order to show the performance of the model. These predictions are made using the backpropagation algorithm which consists of a forward and backward pass. The forward pass uses the given inputs to predict the resulting outputs by evaluating them in the equation:

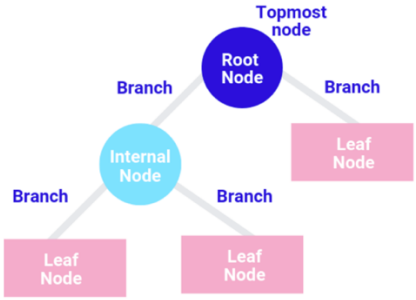
$$\mathbf{x} = \mathbf{f}(\mathbf{s}) = \mathbf{B}\varphi(\mathbf{A}\mathbf{s} + \mathbf{a}) + \mathbf{b}$$

where **s** is the input and **x** is the output. **A** is the weights of the first layer and **a** is the bias of the first layer. **B** is the weights of the second layer and **b** is the bias of the second layer. During the backward pass, the cost function concerning the different parameters are propagated back through the network and the whole process is iterated until the weights have converged.

- The power of these networks arises from their ability to learn training data and how to best relate it to the output that is to be predicted *(profit, performance)*
- They can predict due to their multi-layered architecture with this structure learning to represent different features.

Decision Tree

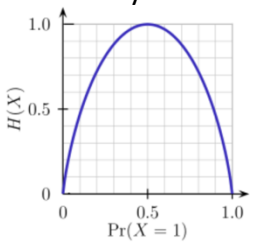
Decision trees are used in building classification models which is useful for fulfilling the requirement of classifying stores into a distinct performance category. Decision trees have a structure that includes nodes, branches, and leaves. The topmost node in the tree is the root node with each internal node defining an attribute, each branch defining a decision and each leaf representing an outcome.



Using this method its decisions can be easily visualised but to do so, two processes must be followed first: Induction and pruning. Decision trees can be prone to overfitting so pruning removes unnecessary structure from the tree and it does this by removing decision nodes starting from the leaf node such that the overall accuracy is not disturbed. This combats this overfitting whilst helping to reduce complexity. Induction describes the process of actually constructing the tree.

Both these processes are commonly carried out by the ID3 algorithm which is a classification algorithm that follows a greedy approach (without backtracking) through selecting attributes with maximum *information gain* or minimum *entropy*.

Entropy is a measure of the amount of uncertainty in a given dataset and is calculated for each attribute with the smallest entropy value used to split the dataset. The higher the entropy the harder it becomes to draw any conclusions from the information.



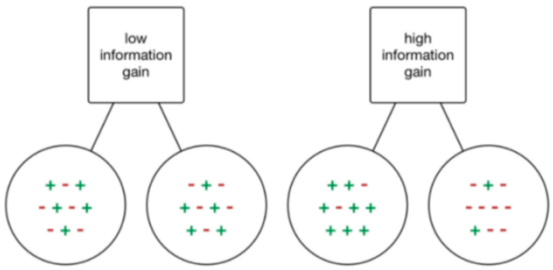
From the above graph, it is seen that entropy is zero when the probability is either 0 or 1. The entropy is maximum when the probability is 0.5 because it shows perfect randomness in the data and the outcome can't be perfectly determined. Mathematically it is represented as:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Where **S** is the current state and **Pi** is the probability of an event **i** of state **S** or the percentage of class **i** in a node of state **S.**

Information gain explains how much uncertainty was reduced after splitting the dataset and it represents a decrease in entropy.

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$



ID3 picks the root node by calculating the information gain for each attribute and picks the one that removes the most uncertainty. Additional branches are added by applying the same information gain calculations and if all objects in a leaf are the same no more branching is needed. The algorithm continues until all values have been accounted for.

Results

As previously mentioned, the models chosen are trying to achieve two distinctly separate functions. The first model is to use prediction to generate a store's revenue based on its values for each variable in the dataset. With this being a prediction task, the Multilayer Perceptron model is the best-suited candidate in carrying out this requirement. The second model is to classify the set of instances into four nominal categories of performance. Based on these specific requirements both the Multilayer Perceptron and decision tree models were assigned to carry out this requirement.

Revenue Prediction

Inputs = *{Staff, Floor Space, Window Space, Car Park, Demographic Score, Location, 40min Population, 30min Population, 20min Population, 10min Population, Store Age, Clearance Space, Competition Number, Competition Score}*

Output = *{Profit}*

Weka's built-in selection algorithm was implemented on the training dataset to extract a subset of original attributes that have the most significant impact on the model performance. Using features with high correlation whilst at the same time removing features of less relevance improves the accuracy of the model and increases the computational efficiency. It also means reducing the amount of data attributes to be collected in the future which in turn reduces costs to the company.

```
=== Run information ===

Evaluator:    weka.attributeSelection.CfsSubsetEval -P 1 -E 1
Search:       weka.attributeSelection.GreedyStepwise -T -1.7976931348623157E308 -N -1 -num-slots 1
Relation:     storedata-training-weka.filters.unsupervised.attribute.Remove-R16
Instances:    91
Attributes:   15
              Staff
              Floor Space
              Window
              Car park
              Demographic score
              Location
              40min population
              30 min population
              20 min population
              10 min population
              Store age
              Clearance space
              Competition number
              Competition score
              Profit
Evaluation mode:    evaluate on all training data


=== Attribute Selection on all input data ===

Search Method:
        Greedy Stepwise (forwards).
        Start set: no attributes
        Merit of best subset found:    0.649

Attribute Subset Evaluator (supervised, Class (numeric): 15 Profit):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 1,3,4,6,13,14 : 6
                     Staff
                     Window
                     Car park
                     Location
                     Competition number
                     Competition score
```

The algorithm as seen above identified a subset of attributes that are the most relevant in evaluating model performance.

Selected Attributes*: Staff, Window Space, Car Park, Location, Competition Number, Competition Score*

Output: *profit*

However, to use the prediction model to estimate the revenue a store should generate, that keeps the average error size to less than half a million pounds, two values will be used and examined: *Correlation Coefficient* and the *Mean Absolute Error*.

Multilayer Perceptron (MLP)

To evaluate the MLP model performance certain combinations of parameters were tuned to establish the best possible solution from the dataset. The parameters evaluated were the:

- **Number of hidden layers** – Defines how many layers exist between input layers and output layers where neurons take a set of weighted inputs and produce an output through an activation function
- **Learning rate –** This controls how much the weights are adjusted with respect to the loss gradient
- **Momentum –** This value can accelerate the training and learning rate and can help converge the optimisation process

Model Number	Parameters			Training set		Testing set	
	Number of Hidden Layers	Learning Rate	Momentum	Correlation Coefficient	MAE Error (£)	Correlation Coefficient	MAE Error (£)
1	1	0.1	0.2	0.7431	369,364	0.4761	571,328
2	1	0.1	0.4	0.7314	377,332	0.4809	588,371
3	1	0.3	0.2	0.6583	441,504	0.4093	704,749
4	1	0.3	0.4	0.6472	452,071	0.4108	725,169
5	2	0.1	0.2	0.7208	384,191	0.4533	649,396
6	2	0.1	0.4	0.6855	408,013	0.4224	680,988
7	2	0.3	0.2	0.5886	489,030	0.4251	580,314
8	2	0.3	0.4	0.5591	512,591	0.3344	670,090

Performance Classification

Inputs = *{Staff, Floor Space, Window Space, Car Park, Demographic Score, Location, 40min Population, 30min Population, 20min Population, 10min Population, Store Age, Clearance Space, Competition Number, Competition Score}*

Output = *{Performance}*

The same process selection used for prediction is also applied when classifying again using Weka's attribute evaluator. Weka uses the correlation coefficient to determine from all the attributes those with a positive or negative correlation to the performance output.

```
=== Run information ===

Evaluator:    weka.attributeSelection.CorrelationAttributeEval
Search:       weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:     storedata-training-weka.filters.unsupervised.attribute.Remove-R15
Instances:    91
Attributes:   15
              Staff
              Floor Space
              Window
              Car park
              Demographic score
              Location
              40min population
              30 min population
              20 min population
              10 min population
              Store age
              Clearance space
              Competition number
              Competition score
              Performance
Evaluation mode:    evaluate on all training data


=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 15 Performance):
        Correlation Ranking Filter
Ranked attributes:
 0.2343   3 Window
 0.2315   1 Staff
 0.2299   2 Floor Space
 0.2143  14 Competition score
 0.2124   4 Car park
 0.1665  12 Clearance space
 0.1176   6 Location
 0.0986  13 Competition number
 0.0805  11 Store age
 0.0763   7 40min population
 0.0525   5 Demographic score
 0.039    8 30 min population
 0.0359   9 20 min population
 0.0295  10 10 min population
```

From the results above the bottom, seven attributes are to be removed from the subset of high correlating attributes. This allows to better evaluate the model performance leaving:

Selected Attributes*: Staff, Window Space, Floor Space, Car Park, Clearance Space, Location, Competition Score*

Output: *Performance*

For the performance classification model to be evaluated the values of *correctly classified instances* and *incorrectly classified instances* are to be defined.

Multilayer Perception

The same parameters defined and used in the prediction MLP model were again used to evaluate the MLP performance model.

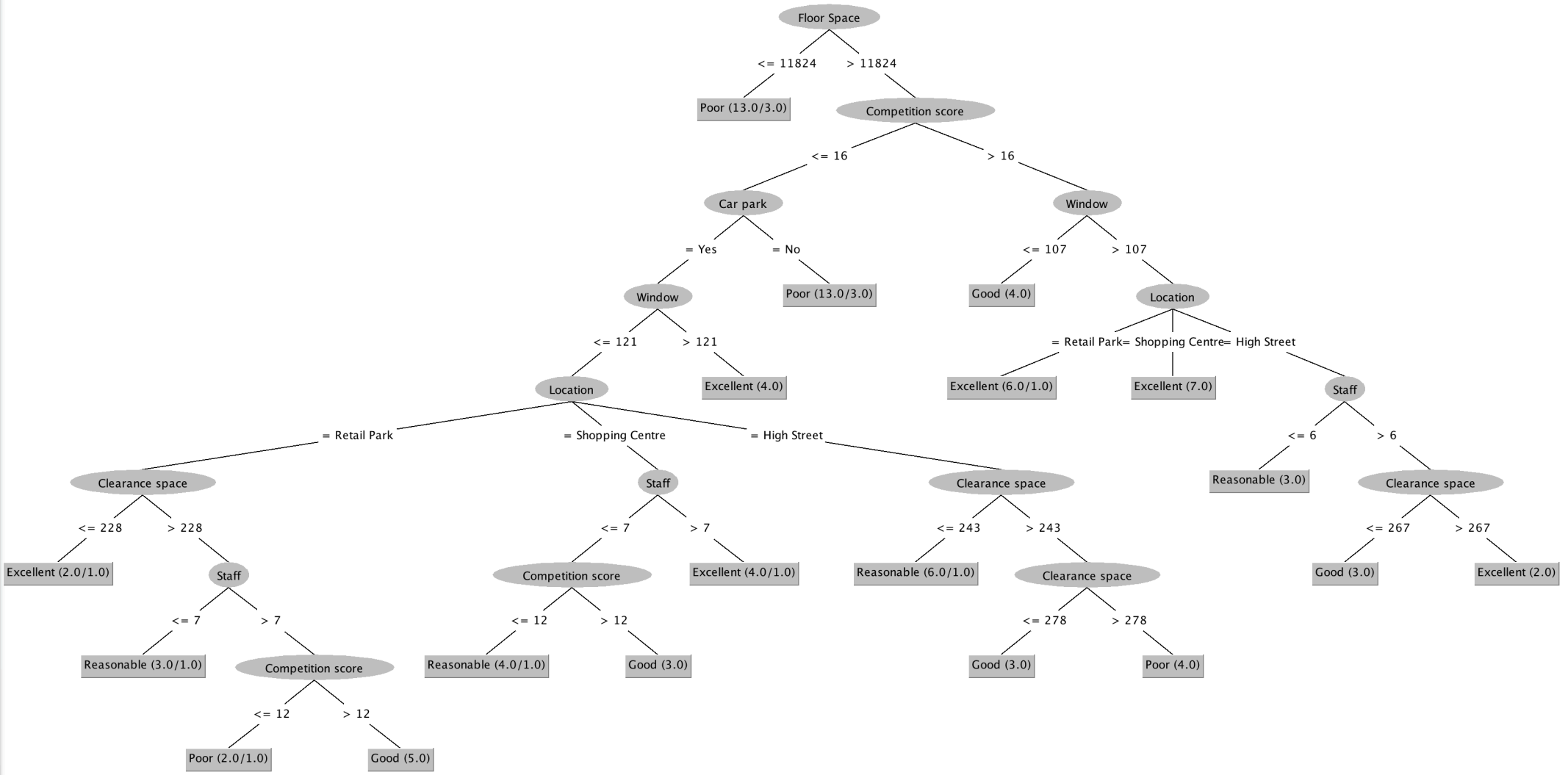
Model Number	Parameters			Training set		Testing set	
	Number of Hidden Layers	Learning Rate	Momentum	Correctly Classified (%)	Incorrectly Classified (%)	Correctly Classified (%)	Incorrectly Classified (%)
1	1	0.1	0.2	47	53	39	61
2	1	0.1	0.4	47	53	39	61
3	1	0.3	0.2	47	53	37	63
4	1	0.3	0.4	49	51	37	63
5	2	0.1	0.2	53	47	37	63
6	2	0.1	0.4	46	54	41	59
7	2	0.3	0.2	46	54	39	61
8	2	0.3	0.4	45	55	39	61

Decision Tree

Similar to MLP to evaluate the decision tree model performance combinations of parameters are to be tuned to establish the best possible solution from the subset. The parameters used to achieve this were the:

- **Confidence Factor** – This is used for pruning in which a smaller value, for example, will incur more pruning.
- **minNumObj -** The minimum number of instances per leaf

Model Number	Parameters		Training set		Testing set	
	Confidence Factor	minNumObj	Correctly Classified (%)	Incorrectly Classified (%)	Correctly Classified (%)	Incorrectly Classified (%)
1	0.1	1	45	55	44	56
2	0.2	1	46	54	44	56
3	0.3	1	44	56	46	54
4	0.4	1	44	56	46	54
5	0.1	2	38	62	44	56
6	0.2	2	46	54	41	58
7	0.3	2	47	53	39	61
8	0.4	2	47	53	39	61


Floor Space
<= 11824
> 11824
Poor (13.0/3.0)
Competition score
<= 16
> 16
Car park
Window
= Yes
= No
<= 107
> 107
Window
Poor (13.0/3.0)
Good (4.0)
Location
<= 121
> 121
= Retail Park
= Shopping Centre
= High Street
Excellent (4.0)
Excellent (6.0/1.0)
Excellent (7.0)
Staff
Location
<= 6
> 6
= Retail Park
= Shopping Centre
= High Street
Reasonable (3.0)
Clearance space
Clearance space
Staff
Clearance space
<= 228
> 228
<= 7
> 7
<= 243
> 243
<= 267
> 267
Excellent (2.0/1.0)
Staff
Competition score
Excellent (4.0/1.0)
Reasonable (6.0/1.0)
Clearance space
Good (3.0)
Excellent (2.0)
<= 7
> 7
<= 12
> 12
<= 278
> 278
Reasonable (3.0/1.0)
Competition score
Reasonable (4.0/1.0)
Good (3.0)
Good (3.0)
Poor (4.0)
<= 12
> 12
Poor (2.0/1.0)
Good (5.0)

Recommendations

The models used in this report can be used to either predict revenue or estimate a classification performance category for a given store and its features. Based on their performances two models are suggested to best fulfil each of two tasks.

For the prediction requirement, the multilayer perceptron model was trained and evaluated to produce the best possible solution. The findings suggest Model number 1 performed at a higher level when compared with the other MLP models. It provided the lowest mean absolute error when the training and test set values were averaged: £470,364. This was below the £500,000 cut-off value provided in the brief and is therefore recommended to be implemented by the company with it being the optimal predictor.

For the classification task, both the multilayer perceptron and decision tree models were trained and evaluated for comparison. The final model that was shown to perform best was the decision tree model number 2. It displayed an average classification accuracy of 45% and will be able to best classify a given store into a performance category and should be implemented by the company to help them understand why a store does better or worse than another store.

With regards to future data collection of store features as can be seen from the data preparation and model implementation several variables should not be paid for:

- Country
- Manager Name
- Store ID
- Town
- Store Age
- Demographic Score

Additionally, all population data should also be excluded from future data collection which should come as a significant saving to the company. The collection of population data as outlined in the brief as some of the most expensive to collect but as can be seen in both the prediction and classification models it was never included in the subset of high correlating variables. Specifically, in the classification attribute evaluation, the population variables performed the worst in correlating with a store's performance highlighting their waste of company resources.