Applied Statistics and Data Mining

Using SAS and R

Thomas Madeley

@00291471

07411988329

t.madeley@edu.salford.ac.uk

MSc Data Science

School of Computing Science and Engineering

University of Salford

January 2021

Table of Contents

Part 1: Classification: Accurately Diagnosing Breast Cancers Using Machine Learning Models

Abstract

Cancer is a leading cause of death in the United Kingdom and across the world. 50% of people in the United Kingdom will be diagnosed with some form of cancer in their lifetime. Among those, breast cancer (BC) is the most common in the United Kingdom with 55,200 cases every year making up 15% of all new cancer cases (Breast cancer statistics, 2020). As with all cancers, early diagnosis is strongly linked to survival rates and therefore creating tools that may increase diagnosis speed may save lives. This paper explores a labelled dataset using a variety of supervised machine learning methodologies with cross validation to assess whether established machine learning models can be used to accurately diagnose breast cancers from digital analysis of cell nuclei.

Introduction

As early diagnosis is closely linked to patient prognosis, it is important to look at ways of streamlining the diagnosis process. The current process involves at least three physical consultations with a physician, the final one being a biopsy: Tissue samples taken from the tumour and then analysed (Breast cancer statistics, 2020). One way to streamline the diagnosis is to reduce total physician hours in the diagnostic process by utilising machine learning to classify biopsy results. Exploratory analysis was performed to determine features of predictive value. Secondly, several machine learning models were deployed to discover the most precise methodology. Finally, k-fold cross validation was performed to verify the findings. In this paper binary classification supervised machine learning methodologies will be used to create an accurate model that can diagnose breast cancers tissues as malignant or benign (Cancerous or non-cancerous) that will be of real-world utility to physicians and clinicians.

What are binary classification supervised machine learning methodologies?

In the data science field, classification refers to the categorisation of data points into a given number of classes. This paper explores binary classification: categorising the data into two classes. Classifications are made by machine learning models. These models employ different mathematical strategies to make the classifications after being trained on, or learning from, input features. An input feature is an individual measurable property of the item being studied. Each entry, or row of the dataset may contain many input features upon which the model may be trained. Classification model training requires labelled data: supervision (Suthaharan, 2016). This means a target outcome,

or classification of each data entry is known: Malignant or Benign in this instance. Most classification methodologies, including the ones used in this paper, follow the same implementation procedure:

- Partition the dataset into training and testing sets.
- Initialise and train the classifier model with the training data with labels
- Use the model to predict the labels of the test set data
- Compare the predictions of the model to the real labels of the test set data

Dataset and Tools

The dataset used for this paper comes from the UCI machine learning repository as is from a study originally conducted by the University of Wisconsin (Street, Wolberg, Mangasarian, 1993). The dataset contains 569 records with 32 real value features. Each record is a digital analysis of an image taken of cell nuclei present in a sample of breast tissue mass taken from a patient with a suspicious tumour. Each record is labelled with a diagnosis of either malignant or benign. All of the features are numeric except Diagnosis which is a character format.

ID	Patient ID Number
Diagnosis	Malignant or Benign
Radius	Mean distance from center to points on perimeter
Texture	Standard deviation of gray-scale values
Perimeter	
Area	
Smoothness	Local variation in radius lengths
Compactness	(Perimeter^2/ Area- 1)
Concavity	Severity of concave portions on the contour
Concave Points	Number of concave points on the contour
Symmetry	
Fractal Dimensio	Coastline approximation - 1

Figure 1 - Meta Data

The key tools used in this paper are RStudio v1.3.1093 with R v4.0.3 and SAS Enterprise Miner Workstation v14.3. Within RStudio the key libraries used were Caret v6.0-86m, rpart v4.1-15, LiblineaR v2.10-8, kernlab v0.9-29 and e1071 v 1.7-4 for machine learning functions, ggplot2 v 3.3.2 was used to create all plots and graphs and dbplyr v2.0 was used for data wrangling and manipulation. SAS Enterprise Miner Workstation was used in its base form.

Data preparation and feature selection

All preparation steps were completed in RStudio. Firstly, the data was checked for missing values and noisy values. The dataset appeared to be complete however there were several entries containing 0 values for features relating to concavity. It was considered that these values may be missing or null data. These entries were kept as a 0 value for concavity is relevant and a reliable measurement: 0 concavity implies the nuclei were flat. The diagnosis column was encoded from nominal values: Malignant (M) and Benign (B) into binary values, M to 1 and B to 0 and the prevalence rate of Malignant and Benign tumours plotted. The prevalence rate of malignant tumours is 37%. The

distributions of all features were plotted, and the distributions are generally normal, with some having a significant skew. Boxplots for each feature were also plotted in order to discover any outliers. Upon consideration outliers were retained. Firstly, because the dataset is relatively small, secondly due to the medical nature of the data, extreme values may be indicative and of predictive value. To mitigate the outlier's effects on the results standardisation will be implemented rather than normalisation as standardisation is more robust to outliers. A correlation matrix was plotted to determine which features were strongly correlated Fig.4. Certain features regarding the outer physical dimensions returned a correlation approaching 1. This is most likely due to features such as perimeter mean and radius mean being mathematically derivative of each other. Features with a correlation approaching 1 were selected to be dropped to reduce the number of features to be used in the machine learning models. as including many highly correlated features can increase the computational workload with minimal effect on model accuracy. The data was then standardised using R's scale function.

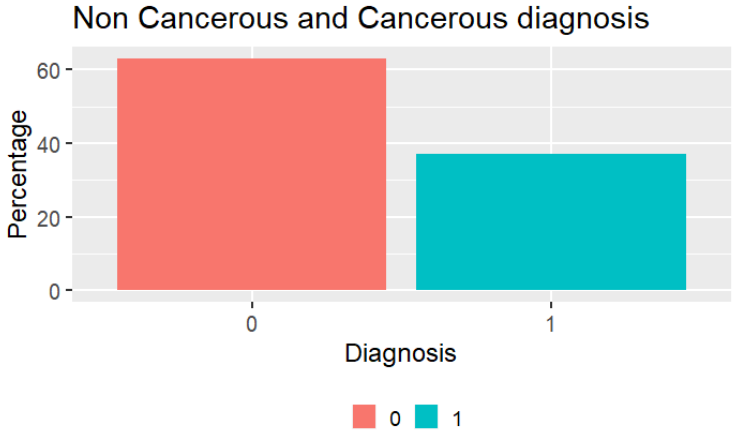


Figure 2 - Diagnosis Distribution

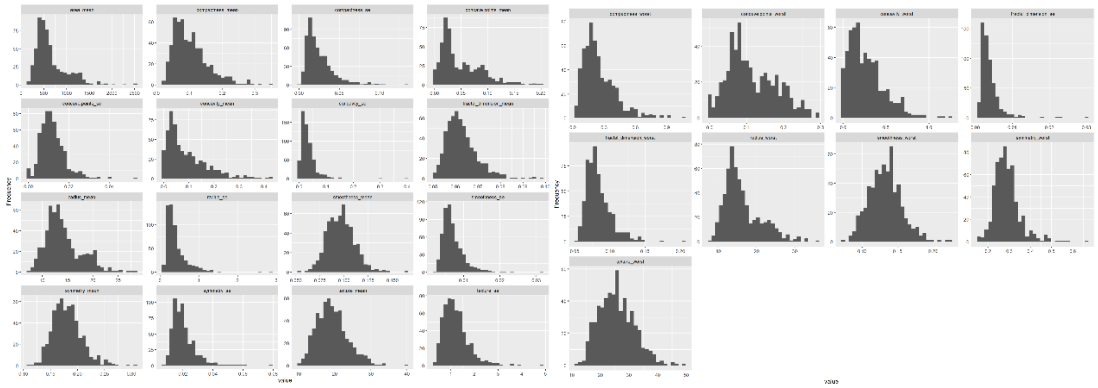


Figure 3 - Variable Distributions

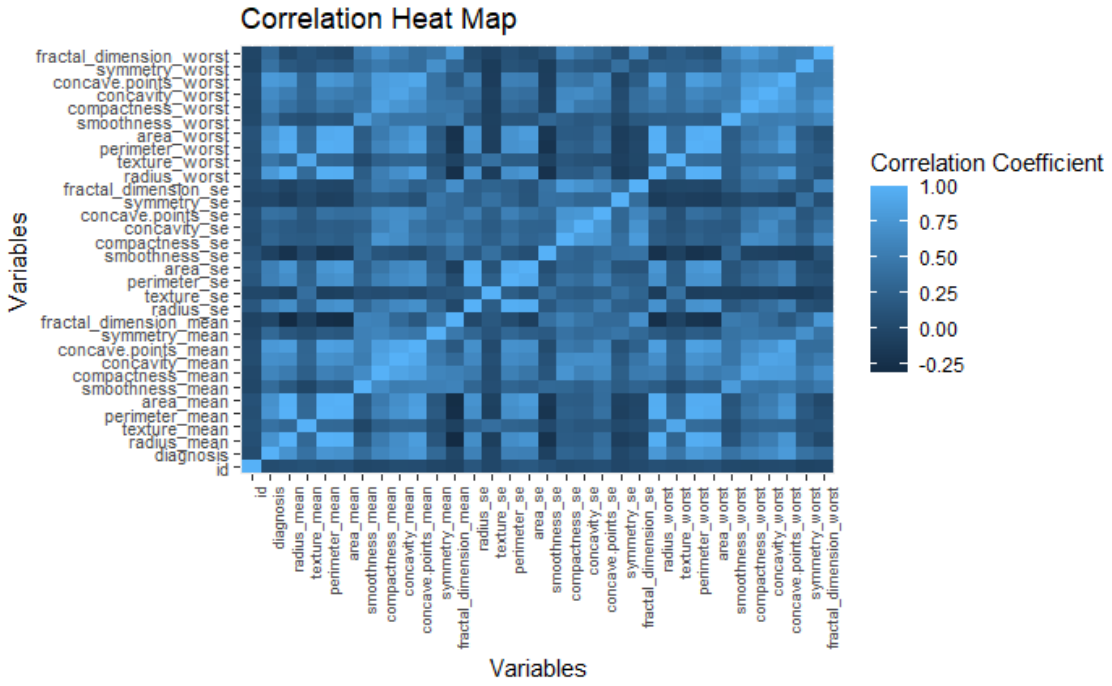


Figure 4 - Correlation Matrix

Models will then be assessed on three parameters: accuracy, sensitivity (sometimes called recall or true positive rate) and specificity (sometimes called selectivity) defined below.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Positives + Total\ Negatives}$$

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

Figure 5 – Equations (Witten and Frank, 2017)

Implementation in R

A random seed was set so that results may be reproducible, then the dataset was divided into a test and training dataset, this will allow for training and testing of the models. This was done using caTool's split function. Next several machine learning models were created and trained with the data from the training set. The model performance was then evaluated using the test set data. The results of each model will be visualised using a confusion matrix plotted with the ggplot2 package.

Logistic Regression

Logistic regression (LR) is a type of binary parametric classification model. LR models require any number of input features with labels for training and outputs a categorical prediction for new input features. In this case the input features are the measured parameters of the cell nuclei and the LR model's output is a probability of diagnosis of either cancerous or non-cancerous. LR model's map the data onto a sigmoid function, which is an 'S' shaped curve whose value approaches 0 and 1 at either end to calculate probabilities. The probabilities are then rounded to 0 or 1 in order to be interpreted as a classification or diagnosis (Hilbe, 2009).

We first implemented the LR using R's 'General Linear Model' (glm) function to create a LR classifier object. We used the predict function on the LR classifier which returns a list of probability of classification between 0 and 1 which were then rounded. Using 'MLMetrics' accuracy function the accuracy value was stored for later comparison with other models. Additionally, Caret's confusionMatrix function was used to get further statistics from the model including sensitivity and specificity.

K Nearest Neighbours

KNN uses Euclidean, Manhattan or Minkowski distances to group similar data points together and make predictions as to the class of new data points. These physical distances can be mapped as a decision boundary. K refers to the number of neighbours specified for the algorithm to consider when classifying a new target data point. Tuning K values is important in order to achieve a stable model: the larger K values rely on many neighbours to make a prediction will give more stable predictions due to averaging a larger number of distances. This continues up to a point where accuracy decreases due to so many distances being considered that points across decision boundaries are considered. Inversely, decreasing the value of K towards 1, the algorithm only considers a small number of neighbours. Any target points close to a decision boundary's single nearest neighbour may be of a different class and the target may be misclassified. In this analysis the 'class' library was used to implement KNN using Euclidean disances and values between 1 and 10 were used for K . Caret's confusionMatrix function was used to calculate accuracy, sensitivity and specificity.

Support Vector Machine

Support vector machine (SVM) is more recent classification model, originally proposed in the 1960's and revised in the 1990's by Vladimir Vapnik, and have only recently gained popularity in the machine learning field (Boser, Guyon, Vapnik , 1992). The SVM model creates decision boundaries using a maximum margin hyperplane. SVM tries to find the optimal decision boundary between by plotting a line, or where not linearly separable, a hyperplane between the two nearest points of different classes. The line, or hyperplane, must be equidistant from both points to maximise the

margin, these points are known as support vectors. When not linearly separable SVM uses various mapping functions to map the data into another dimension where it may be linearly separated by a hyperplane (Kowalczyk, 2018). That SVM uses the boundary points of two classes, or extreme cases, differentiates SVM from other classification methods. The algorithm was implemented using e1071 library's svm function. Both a linear kernel and a radial kernel were used independently. Caret's confusionMatrix function was used to calculate accuracy, sensitivity, and specificity.

Decision Tree Classification

Decision Trees are a very simple algorithms that can be used for both classification and regression. Decision Tree Classification (DTC) iteratively splits the data to minimise informational entropy. These splits divide the data into leaves, the final split returns terminal leaves that will contain the classification results (Suthaharan, 2016). The DTC was implemented using the rpart package's rpart function. The model was tuned using the minsplit and minbucket hyperparameters, defining the minimum number of observations that in a leaf to be split and the minimum number of observations in a terminal leaf respectively. These parameters how deep the tree can grow, having a reasonably large minsplit and minbucket can help prevent overfitting of the model to the training data. Optimum values were found to be 18 for minsplit and 4 for minbucket.

Repeated K Fold Cross Validation

The accuracy of above models is only validated using the small test set (20%). As a random seed was set, this is just one permutation of splitting the data into training and test sets, how can one be sure that this model is not overfitted to this test set? Unfortunately, there is no further data to test the model on. It was decided that, in order to access the reliability of the results, that cross validation would be performed. K Fold Cross Validation (KFCV) randomly divides the dataset in a set number of parts or *folds* (K) of approximately equal number of rows. One of these folds is retained and used to test the model later. The remaining folds are used to train the specified model, the model is then tested, and an accuracy score produced. This process is then repeated a specified number of times, folding the data randomly and randomly selecting a fold with which to test the model (Rodriguez, Perez & Lozano, 2010). Additionally, KFCV in R can be used to tune hyperparameters as well as pre-processing. A range of hyperparameters may be specified and will be iteratively implemented.

KFCV was implemented on the SVM Radial Kernel Model using the Caret library's createMultifFolds, trainControl and train functions. To increase the amount of data available to the cross validator, the entire dataset was used rather than the smaller training set. The number of folds (K) was set to 5. As one fold is retained for testing, this would simulate the 80% to 20% train to test split used in the original model. The hyperparameter 'cost' was tested for integers between 1 and 10. The train function also carried out the standardisation of the dataset Cross validation was repeated 10 times for each cost value and the mean accuracy for each cost value reported meaning that in total, the SVM radial kernel model was tested 500 times with different permutations of the dataset. This was

very computationally expensive and even with the small dataset took a long time to complete (Hours). To reduce the time taken to train and test these 500 models, the 'doSnow' library was utilised. RStudio is a single threaded application, meaning it only uses 1 CPU thread to perform calculations. 'doSnow' Allows the user to specify how many threads RStudio will use for a task. Ten threads were used dramatically reducing compute time.

```r
```{r}
library(caret)
library(doSNOW)
library(LiblineaR)
library(kernlab) # for svmRadial


set.seed(123456)
cl<- makeCluster(10, type="SOCK")#Tells R to use 10 cpu threads instead of 1.
registerDoSNOW(cl) # initialises the 10 cpu thread cluster

# k = number of folds
#times = number of repeats
#Creates lists of indexes of rows for the folds
cv_folds_index <- createMultiFolds(as.factor(dataset_dropped$diagnosis), k=5, times = 10)

#specifies parameters for train function
#method = repeated cross validation
#number = number of folds
#repeats = number of repeated validations
#index = indexes used to create the folds
cross_validator_control <- trainControl(method = "repeatedcv", number = 5, repeats = 100, index
= cv_folds_index)


#Uses the trainControl parameters to build model and cross validate
#x= training parameters, labels removed
#y= training labels
#method = model type: SVM Radial Kernel with Cost parameter
#trControl = control parameters specified above
#tunegrid =  values for Cost to be test: from 1 to 10 for 10 values
#preProcess = Scaling, centered
svm_cross_validator <- train(x = dataset_dropped[-1], y = as.factor(dataset_dropped$diagnosis),
method = "svmRadialCost", trControl = cross_validator_control, tuneGrid =
expand.grid(C=seq(1,10, length = 10)), preProcess = c("center", "scale"))


stopCluster(cl) #closes the 10 cpu thread cluster to return to 1 thread



```
```

Figure 6 - K Folder Cross Validation Code Snippet

Results In R

The confusion matrices of each model were analysed to calculate the accuracy, sensitivity and specificity of each model so that the optimum model could be discovered. The confusion matrix categorises the model predictions into True Positives (TP),True Negatives(TN), False positives(FP) and False Negatives(FN). In this context a TP refers to malignant nuclei correctly classified and malignant, TN refers to benign nuclei correctly classified as benign. FP refers to benign nuclei classified as malignant; FN refers to malignant nuclei classified as benign. In order to offer the best care to patients it can be deduced that TP must be maximised and FN minimised and therefore accuracy and sensitivity must be maximised as per formula 1 and 2. In addition, Sensitivity must be considered as important, if not more important than accuracy due the dangers of FN relative to the minor dangers of FP: a person with cancer going undiagnosed vs a person being referred for treatment when they do not need it. A perfect model would have an both an accuracy and sensitivity approaching 100%.

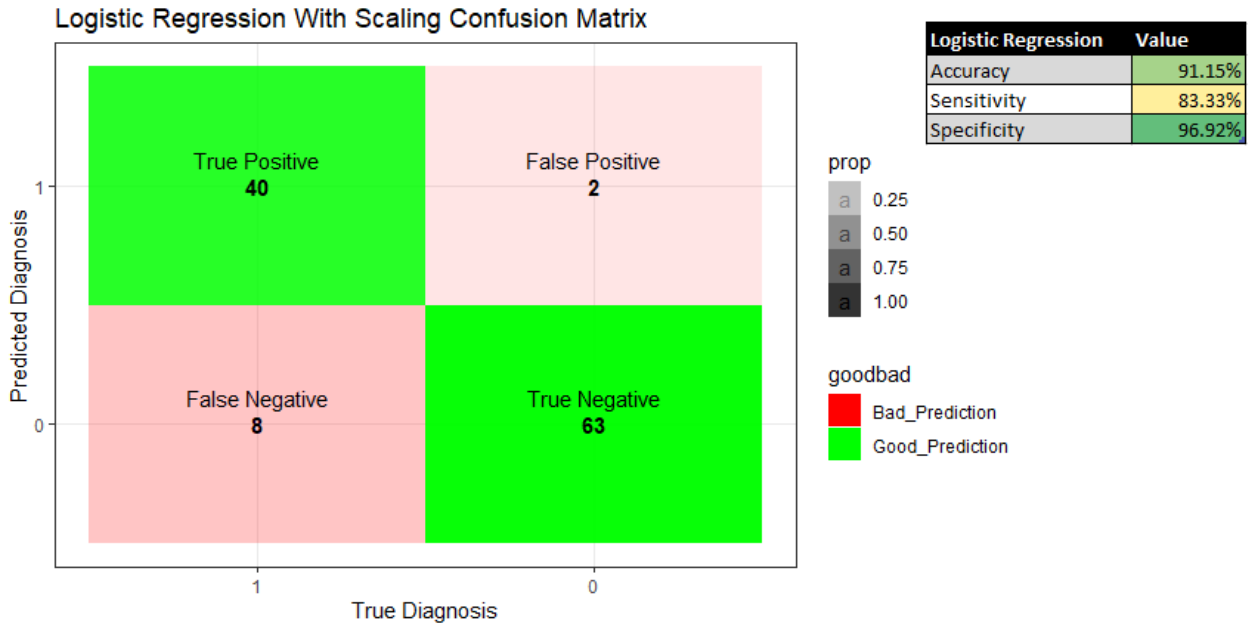


Figure 7 - Logistic Regression Confusion Matrix

LR returned a 91.15% accuracy with 10 misclassifications in total, with most of the misclassifications being false negatives: Cases of cancer that were not correctly diagnosed. This resulted in a low Sensitivity.

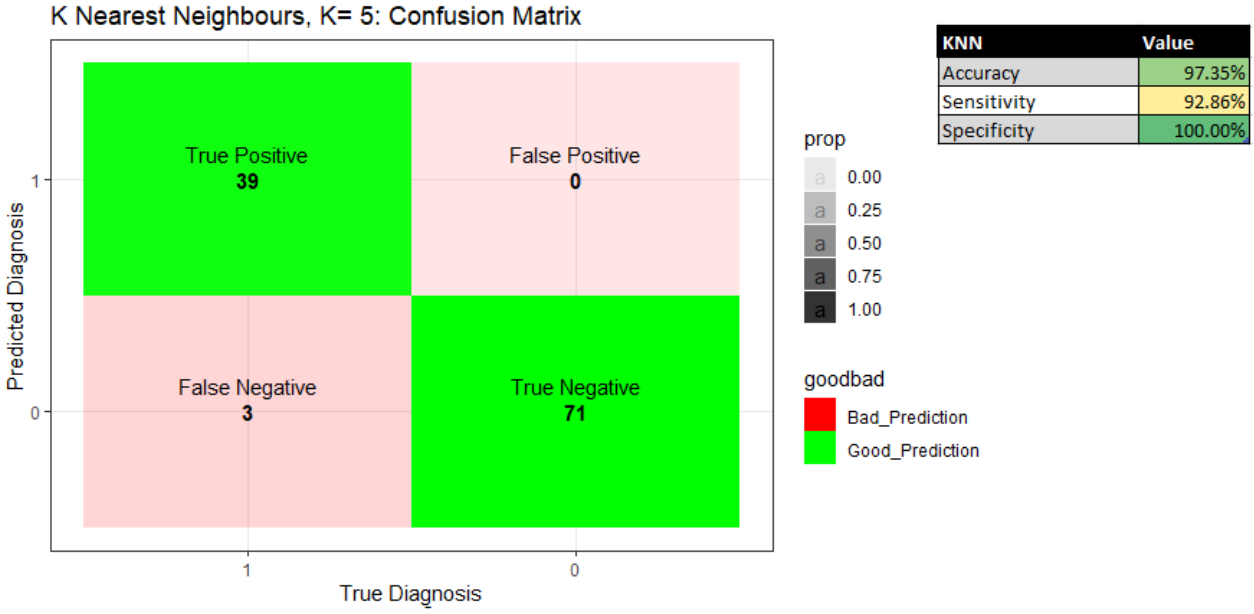


Figure 8 - KNN Confusion Matrix

KNN returned a higher overall accuracy at 97.35% than LR with a K value of 5. There were no false positive results and therefore specificity was 100%. There were only 3 misclassifications but unfortunately these were all false negatives.

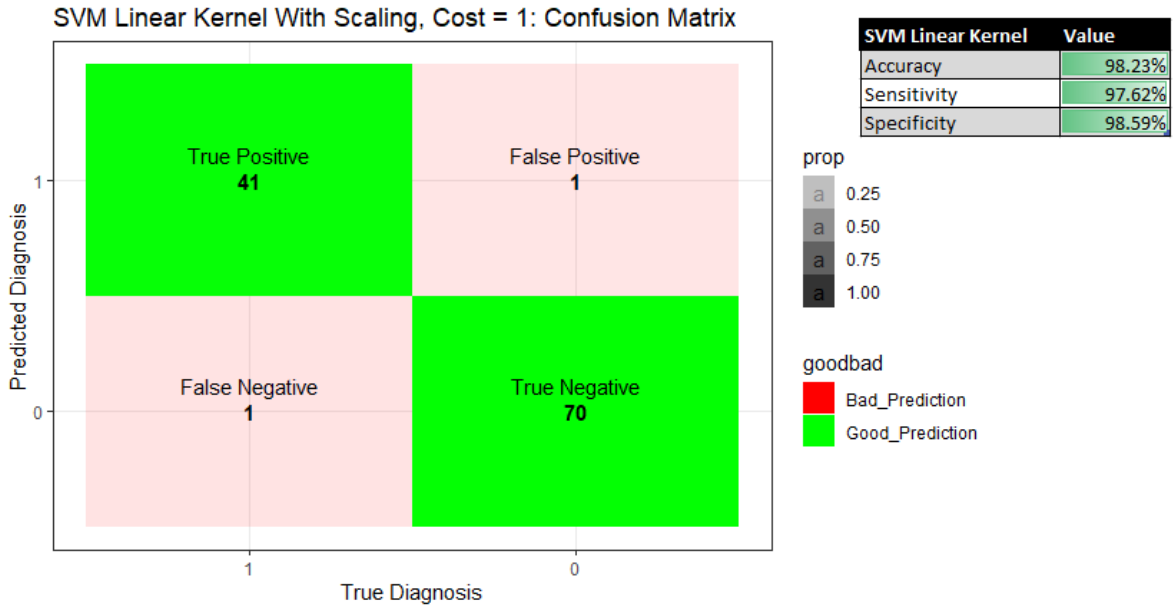


Figure 9 - SVM Linear Confusion Matrix

SVM with the linear kernel returned excellent results with only two misclassifications. The Cost parameter was tuned by iteratively trying all values between 1 and 10. 1 returned the highest accuracy.

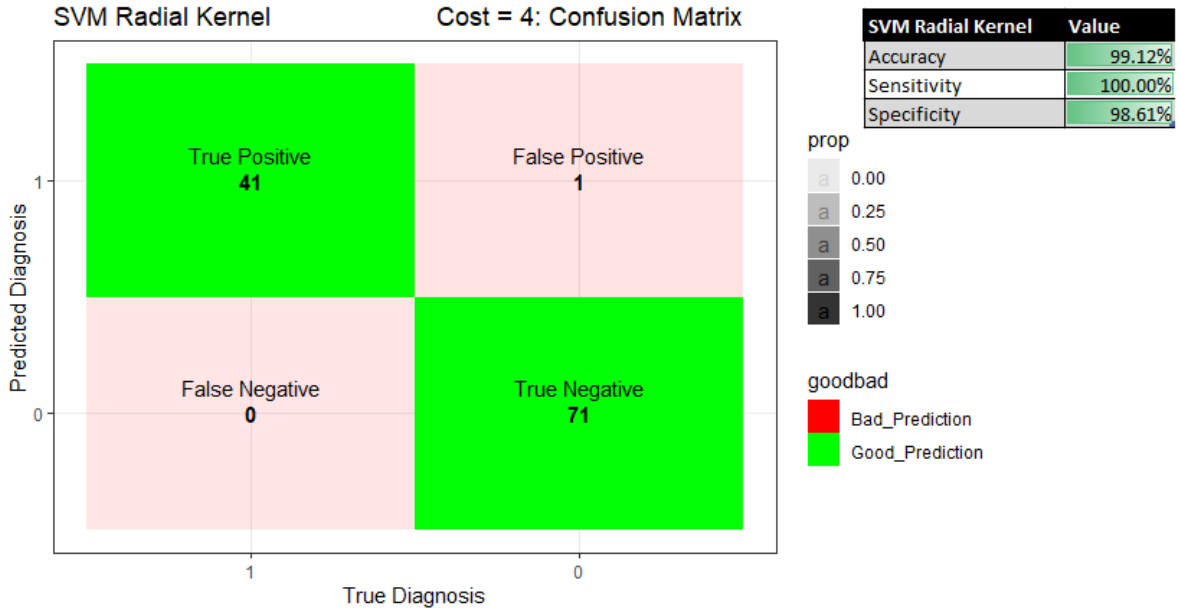


Figure 10 - SVM Radial Confusion Matrix

SVM with the radial kernel was the most accurate model returning only 1 misclassification: a false positive. This model had 100% sensitivity: all positive cancer cases were diagnosed correctly.

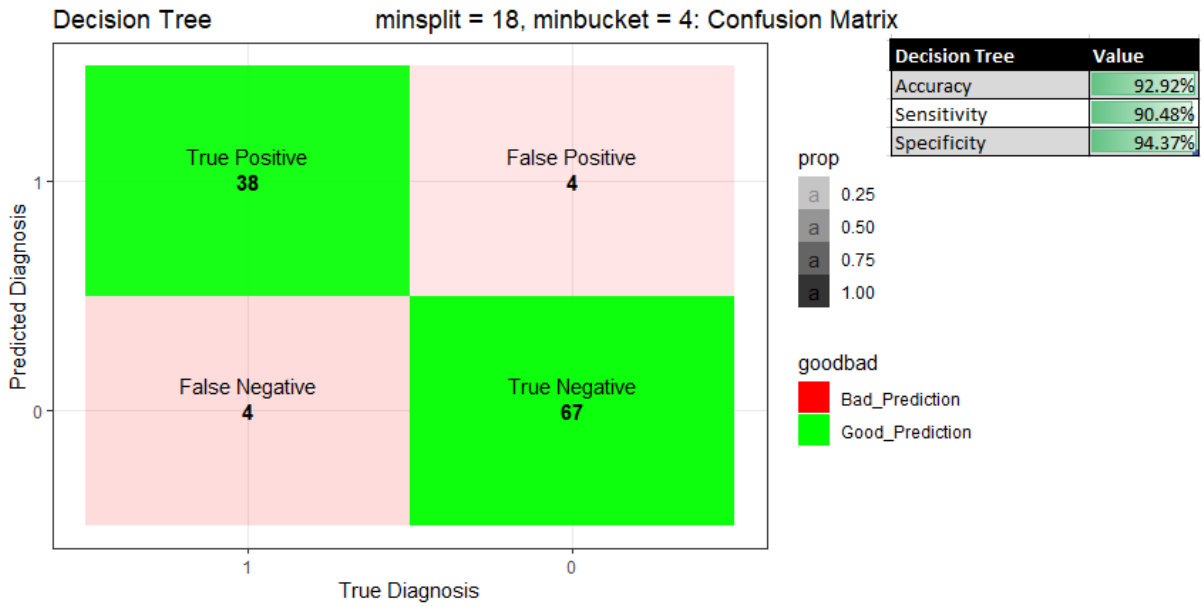


Figure 11- Decision Tree Confusion Matrix

DTC performed consistently across all three parameters but was ultimately outperformed by the two SVM models.

The SVM radial kernel had the highest accuracy with a score of 99.12% which equates to only one misclassification. As this misclassification was a false positive, the SVM radial scored 100% sensitivity meaning that no malignant nuclei were misdiagnosed: All patients that have cancer would be referred for treatment and only 1 patient with a benign tumour would, incorrectly, be referred for treatment. It was also noted that SVM linear kernel's results were similar with 98.23% accuracy and 97.62% sensitivity, respectively. In addition, while KNN achieved high accuracy at 97.35%, the sensitivity was very low at 92.86% making it unsuitable for this purpose. Conversely the KNN specificity was 100% meaning KNN may have applications when a FP result may be dangerous.

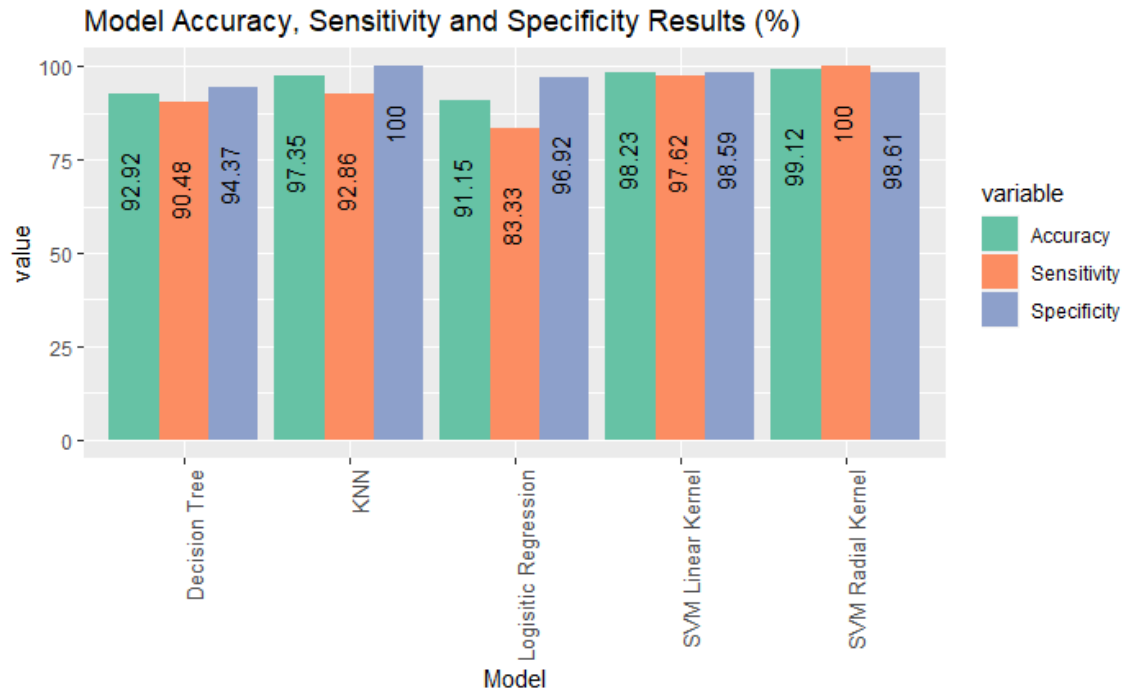


Figure 12 - Model Statistics Plot

SVM radial kernel produced the highest accuracy and sensitivity and is therefore the most successful model at diagnosing breast cancers in this dataset. The accuracy was high, and it was considered that perhaps the model was overfitted to this training set. It was decided that further testing of the model was required. KFCV's implemented in order to randomise the testing and training of the model to reproduce the accuracy seen in the initial test and training set.

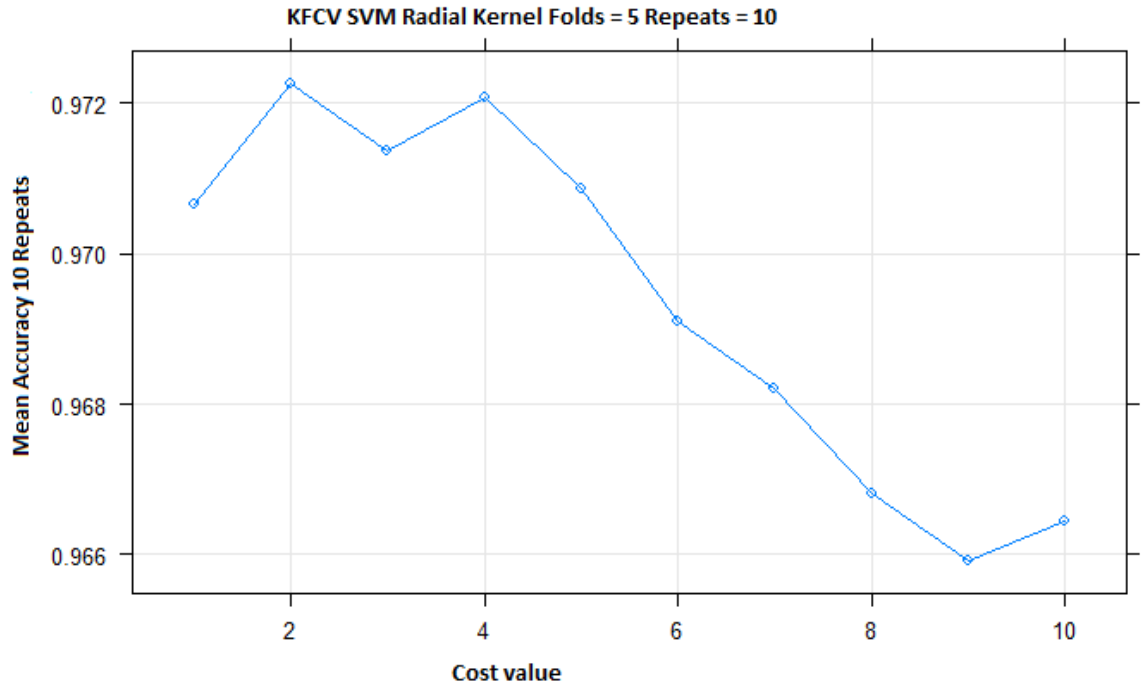


Figure 13 - Cross Validation Mean Accuracy by Cost Value

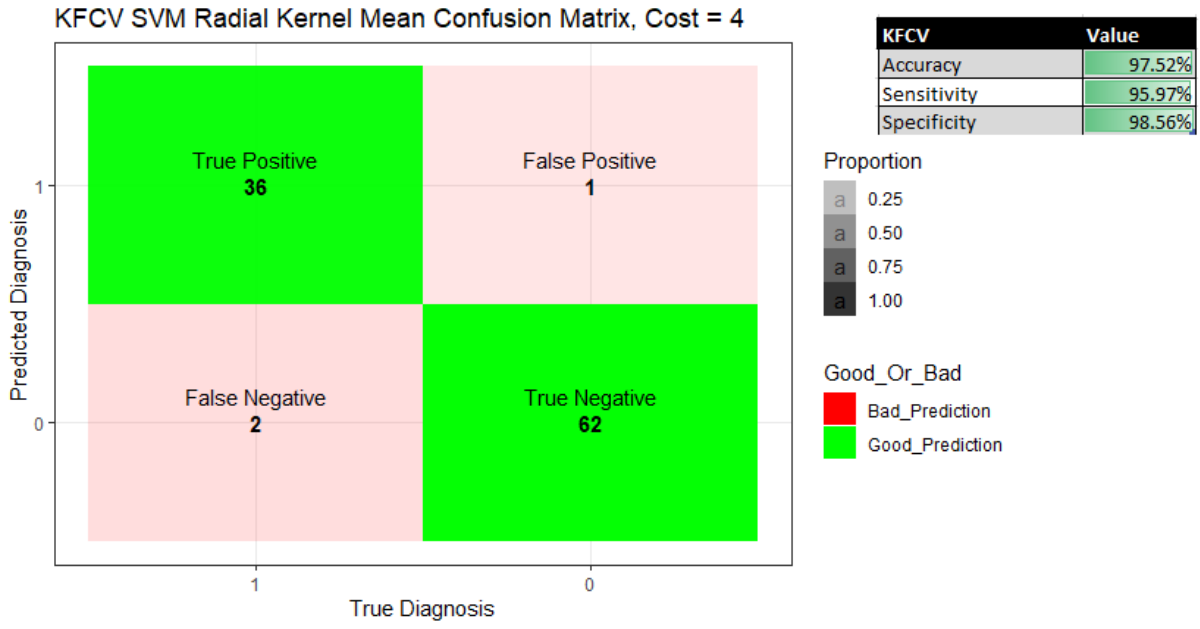


Figure 14 - Cross Validation Mean Accuracy Cost = 2

When model training and testing repeated that a cost value of 2 results in the highest accuracy. The mean accuracy for the best cost value is only 97. 22%. This means that the SVM radial kernel was slightly overfitted to the initial test and training set: corresponded too exactly to that particular set of data and therefore may have failed to classify future data with the same accuracy reliably. The KFCV results show that the SVM radial kernel model can reliably achieve 97.22% accuracy on new data.

SAS Enterprise Miner Implementation

Initially a SAS diagram was created. The file import node was used to load the dataset. Next the data partition node was used to separate the data into training and validation sets, the ratio was 80% and 20% respectively. The partition method was set to simple random rather than the SAS default, stratified. Stratification would split the data into proportionally equal groups based on the target variables, 37% M and 63% B. It was decided that a simple random split would better access the model as stratified random partitions from this population may not be indicative of the populations in future samples. The random seed was set in the data partition node. Using the transform variables node the features were standardised. The standardised data was then analysed by the stat explore note. Each variables' worth was plotted and variables with a worth below 0.1 were dropped.

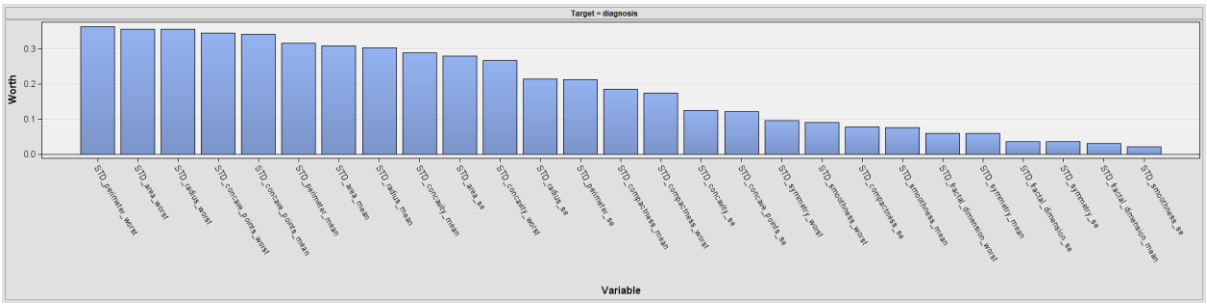


Figure 15 - SAS Variable Worth Plot

Next a control point was used on the process flow in order to feed the data into multiple model nodes. The relevant nodes to implement logistic regression, KNN, decision tree and SVM linear kernel and SVM radial kernel were used. Finally, the model nodes were linked to a model comparison node.

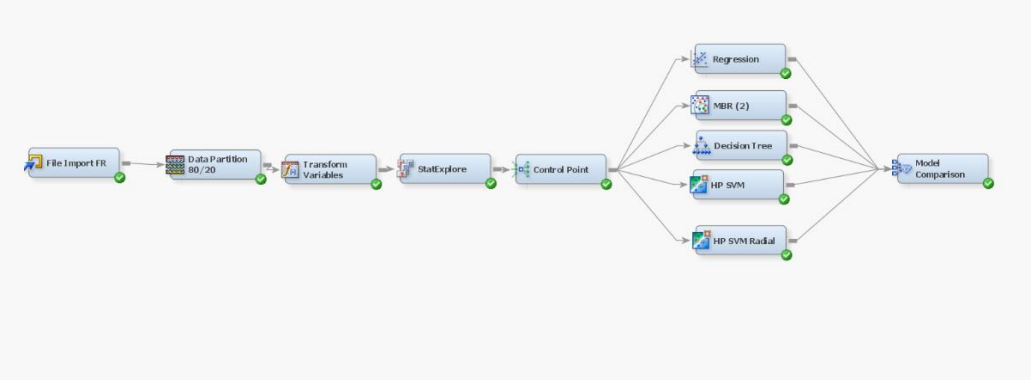


Figure 16 - SAS Process Flow Diagram

Implementing Models in SAS

The regression node was configured to perform logistic regression and the 'link function' parameter was tuned for best accuracy. The memory-based reasoning node (MBR) was used to perform KNN was tuned using the number of neighbours (K) parameter was tuned and it was found that 3 gave the best accuracy. The decision tree node was used to implement the decision tree algorithm. Maximum categorical size, similar to minsplit in R, was tuned for accuracy. Maximum depth was kept to the default value of 10 as the tree consistently did not reach a depth of more than 5 due to pruning. The HP SVM was used twice to implement SVM with both a linear and radial kernel. To create the linear SVM model, 'Optimisation Method' was set to 'Interior Point' and then 'Interior Point Options' were configured to use a linear kernel. To create the radial SVM model 'Optimisation Method' was set to 'Active Set' and 'Active Set Options' set to use the 'Radial Basis Function'. The Penalty (Cost in R) function was tuned for both models for best accuracy.

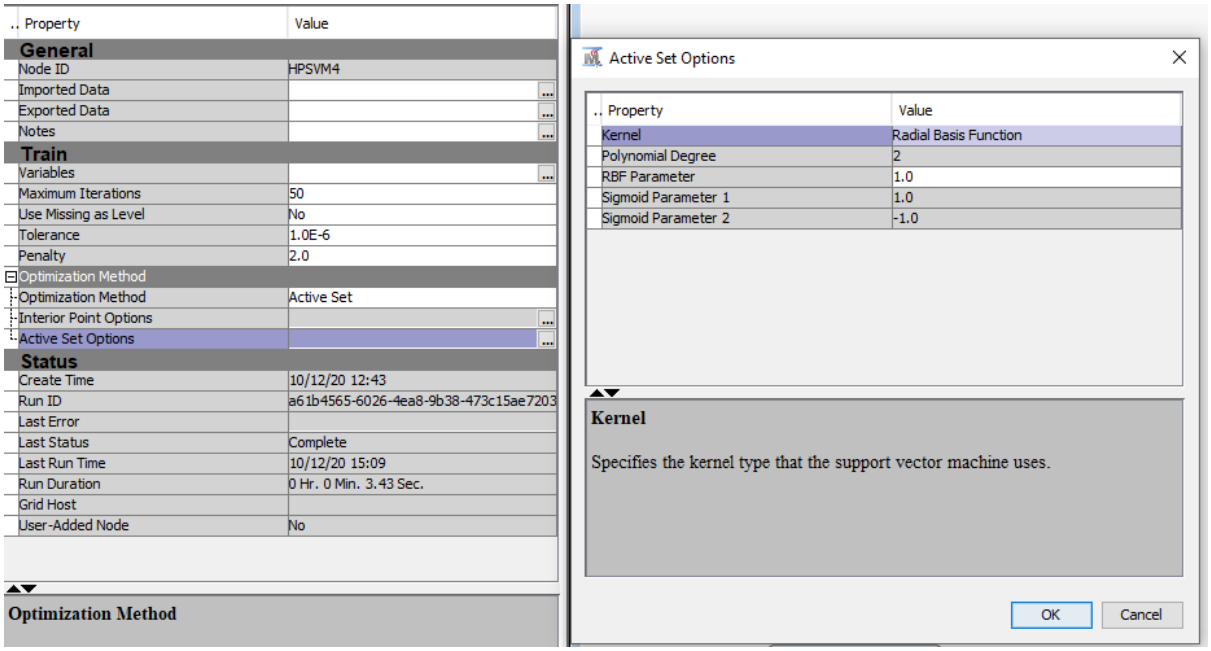


Figure 17 - HPSVM Node Configuration

The 'Model Comparison' node's 'Model Selection' parameters were changed so that the ROC Index of the validation data would be the model selection criteria.

Results in SAS

The Model Comparison node was used to generate a receiver operating characteristics curve (ROC curve). The curve is constructed by plotting sensitivity (true positive rate) over 1 minus specificity (false positive rate). The closer a model plot to the top left of the ROC curve, the better the model. Reaching the top left of the ROC plot would indicate a high sensitivity and a low false positive rate. The straight diagonal line represents the baseline: the outcome that would be expected when binary classification is done using random selection.

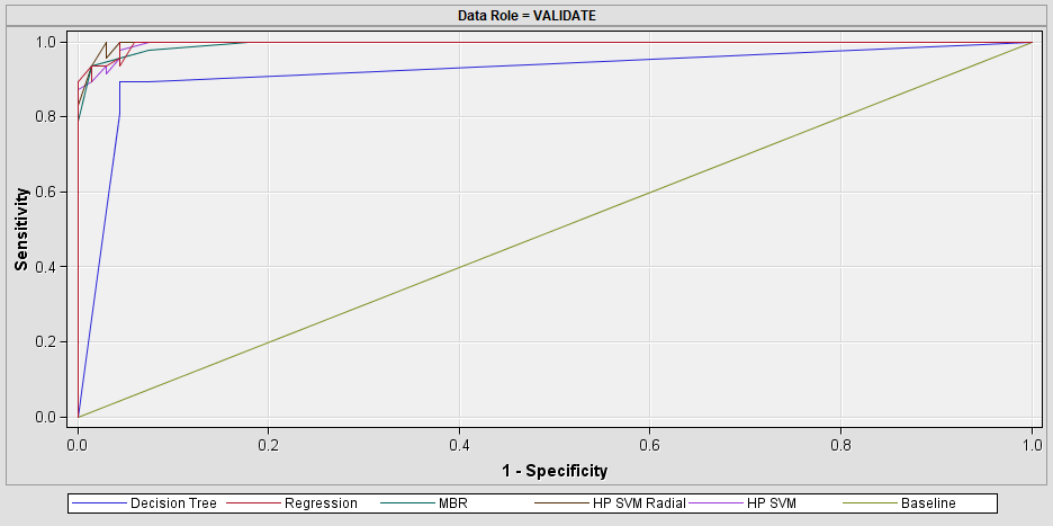


Figure 18 - SAS ROC Curve

All the models performed better than the baseline. DTC performed visibly worse than the other models and is much further from the top left of graph. LR, MBR, SVM linear and SVM radial all performed very similarly and it is very difficult to identify the superior model. As the superior model will be closest to the top left of the chart, it can be deduced that the superior model will have a larger area under its curve. SAS calculates this area under curve (AUC) or ROC index statistic in the 'Fit Statistics' function in the 'Model Comparison' node. The 'Model Comparison' has evaluated the models and selected the SVM Radial model as the optimum. It selected this model as the selection parameter was assigned to ROC index and this model has the highest ROC index. In addition, the SVM radial model had the joint lowest misclassification rate of the validation data.

$$Missclassification\ Rate = 1 - Accuracy$$

$$SVM\ Radial\ Accuracy = 96.5\%$$

Selected Model	Model Description	Target Variable	Selection Criterion: Valid: Roc Index	Valid: Misclassification Rate	Train: Misclassification Rate
Y	HP SVM Radial	diagnosis	0.998	0.035088	0.037363
	Regression	diagnosis	0.997	0.04386	0.028571
	HP SVM	diagnosis	0.996	0.052632	0.057143
	MBR	diagnosis	0.995	0.035088	0.037363
	Decision Tree	diagnosis	0.921	0.070175	0.03956

Figure 19 - SAS Model Results By ROC Index

It should be noted that KNN (MBR) and LR also scored very highly with accuracies of 96.5% and 95.6% respectively. Further analysis of the confusion matrices was conducted to calculate accuracy,

sensitivity and specificity. A classification chart was also plotted in the 'Model Comparison' node using 'data options' to display the classification count rather than percentage.

Logistic Regression SAS	Value	KNN SAS	Value	SVM Linear Kernel SAS	Value	SVM Radial Kerne	Value	Decision Tree SAS	Value
Accuracy	95.60%	Accuracy	96.49%	Accuracy	94.74%	Accuracy	96.49%	Accuracy	92.98%
Sensitivity	95.74%	Sensitivity	93.62%	Sensitivity	91.40%	Sensitivity	93.62%	Sensitivity	89.36%
Specificity	95.52%	Specificity	98.51%	Specificity	97.01%	Specificity	98.51%	Specificity	95.52%

Figure 20 - SAS Model Results

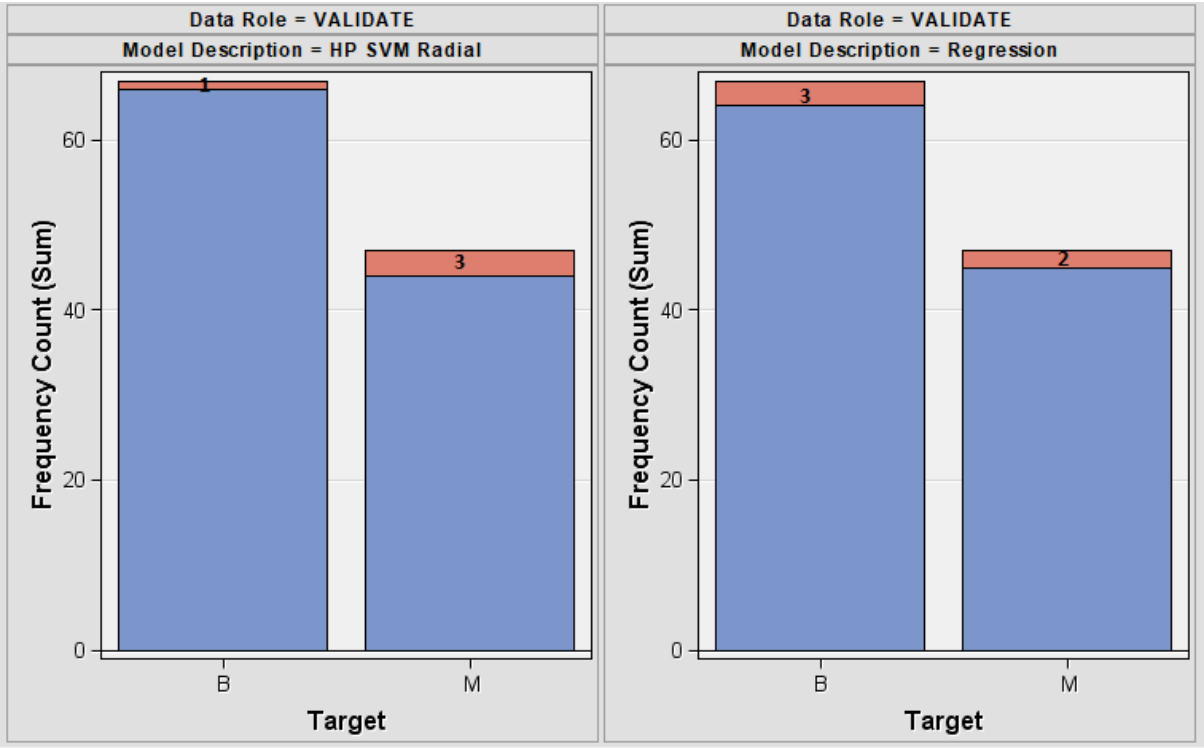


Figure 21 - SAS SVM vs Logistic Regression Plot

It was noted that while SVM radial and KNN scored the highest accuracy and ROC index, LR had a higher sensitivity with a ROC index only very slightly below that of SVM. This is due to LR only classifying 2 FN when SVM radial classified 3 FN. Due to the higher sensitivity it was concluded that LR is the superior model for breast cancer diagnosis despite its lower accuracy as more true cancer cases were identified.

Results Comparison and Conclusion

While there was a difference in the approaches, both SAS and R ultimately concluded that SVM radial kernel was the best model for classifying this dataset. While R's final accuracy was slightly higher than the accuracy reported in SAS, the difference was very small at only 0.7%. Despite this, the conclusion drawn in SAS was ultimately overruled as the LR model had better sensitivity. This result was not consistent with the LR model in R. The differences in LR models across the two platforms is due to a difference in link function. The link function is the mathematical function that

relates an input variable to the linear model. In SAS the best accuracy and sensitivity for LR was achieved with the Probit link function, in R the link function was unchanged and the default function is Logit'. Additionally, the difference could also be due to either the LR model in SAS being over fitted: the model is fitted too closely to the training and test data and therefore accuracy observed may not be consistent when presented with new data. Or the LR model in R may be under fitted: the model is fitted too loosely to the training and test data and therefore accuracy observed may not be consistent when presented with new data.

While the LR model performed well in SAS, the results were not consistent with what was achieved in R. As this SAS LR model was not cross validated, and its performance on new data unknown, it cannot be put forward for future use despite the high sensitivity and accuracy achieved. In conclusion the SVM radial kernel model is recommended for future use in classifying breast cancer data. The SVM radial kernel model performed consistently well across both platforms achieving the highest accuracy on both. SVM radial kernel also achieved the highest sensitivity in R. The R model was also cross validated and can be expected to perform at least 97% accuracy when presented with new data.

Creation of Shiny Dashboard

Classification: Accurately Diagnosing Breast Cancers Using Machine Learning Models (shinyapps.io)

A shiny dashboard was created to display the results of this classification paper (Link Above). The dashboard was created with the shiny, shinydashboard and shinyBS packages. Firstly, the key points an prospective user would want to see were considered. These were deemed to be: The optimal model the paper found, the parameters by which the model was deemed optimal and finally the number of cancer cases not correctly diagnosed. A brief title and subtitle were added to explain the purpose of the dashboard. To display the key points, a 'KPI' banner would be used to display the model chosen, the accuracy of said model and the number of cancer cases misdiagnosed so the user can get this information immediately. This was done using the infoBox function with relevant icons chosen. Secondly, the main scoring parameters of all the models were displayed using a ggplot barchart below the KPI banner using the plotOutput function. As a user may have a specific model of interest, the selectInput function was used to allow the user to select which models they would like to view in the bar chart. A checkbox input was considered, but due to memory limits with the free shinyapps.io accounts, it was not possible. Using the selected function, all models were displayed by default. A tooltip was then added to the bar chart using the bsTooltip function, to give a brief explanation of how to use the chart.

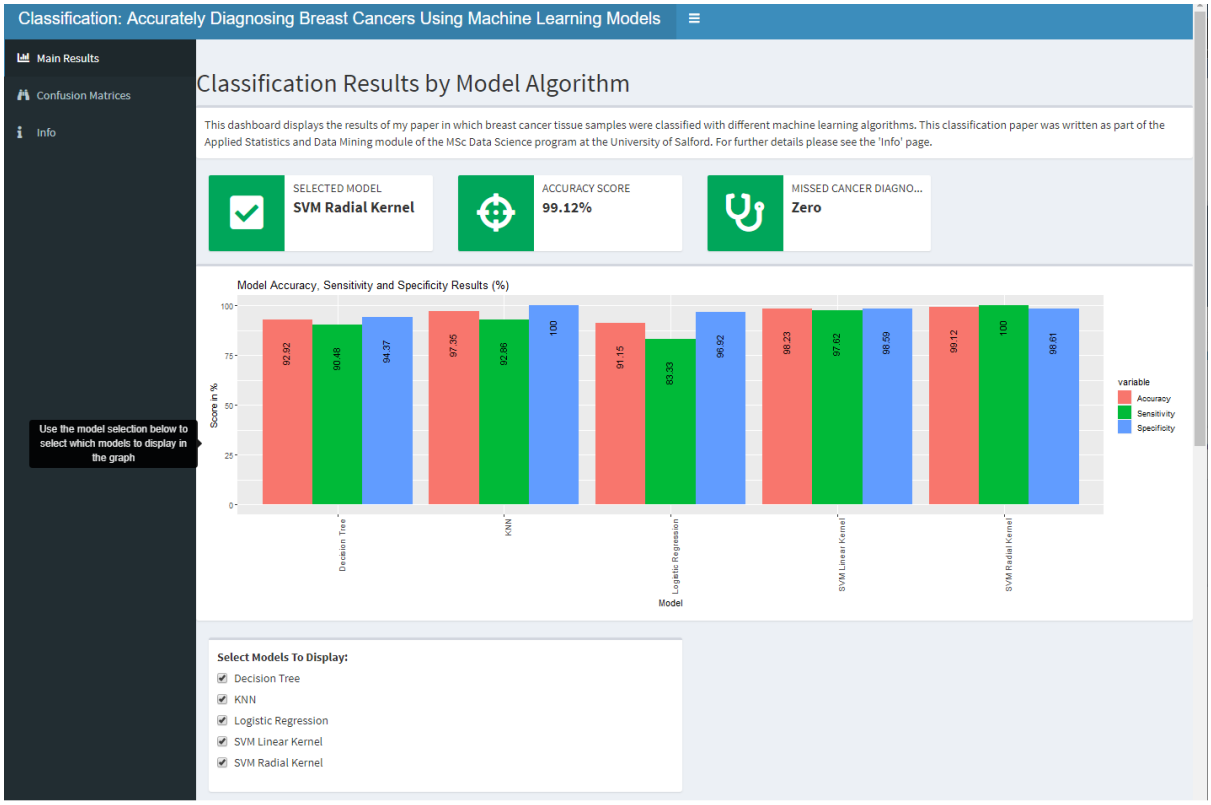


Figure 22 - Shiny Dashboard Homepage

Below this, as the user scrolls down, a brief explanation of the dataset used was displayed before an interactive view of the dataframe allowing a user to explore the data.
A second page was added to the dashboard using the menuItem function. This page displays the confusion matrices for all the models to give further context to the main results barchart. To reduce clutter, the confusion matrices were made collapsible, with the default state being collapsed. This meant that the user can view and hide the matrices at will. Each matrix was given a tooltip with a brief explanation of the results.

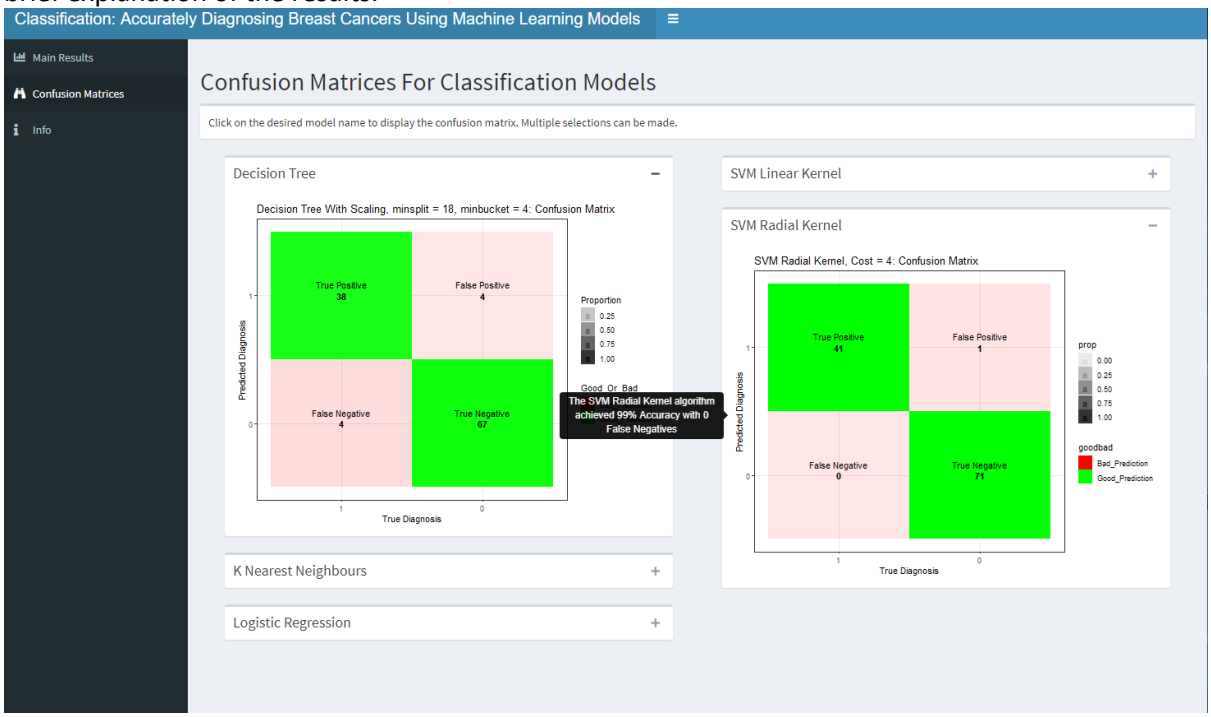


Figure 23 - Second Dashboard Page

Finally, a third page was added to display further information about the paper, the paper's abstract, where the dataset could be retrieved and where the paper could be read. These text boxes were also made collapsible so that the user could view only relevant information to their investigation.

Several issues were encountered when uploading the dashboard to shinyapps.io. Small issues such as using relative paths were resolved quickly and easily. The most severe issue was the 1 Gigabyte memory limit of the free shinyapps.io accounts, which the dashboard exceeded. To reduce the memory usage of the dashboard, the R code was optimised. Intermediate dataframes were replaced with piped functions where possible and this reduced the memory usage to acceptable limits.

REFERENCES:

W. Nick Street, W. H. Wolberg, and O. L. Mangasarian "*Nuclear feature extraction for breast tumor diagnosis*", Proc. SPIE 1905, Biomedical Image Processing and Biomedical Visualization, (29 July 1993)

Cancer Research UK. 2020. *Breast Cancer Statistics*. [online] Available at: <https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/breast-cancer#heading-Zero> [Accessed 09 December 2020].

Boser, B. E., I. Guyon, and V. Vapnik (1992). *A training algorithm for optimal margin classifiers*. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, (pp 144 -152).

Vapnik, V. (2010). *The nature of statistical learning theory*. New York: Springer.

Kowalczyk, A. (2018). *Support Vector Machines Succinctly* (pp. 32 - 45). Syncfusion Books.

Hilbe, J. (2009). *Logistic regression models* (1st ed., pp. 353 - 410). CRC Press LLC.

Suthaharan, S. (2016). *Machine learning models and algorithms for big data classification* (pp.160-162) Springer International Publishing.

Suthaharan, S. (2016). *Machine learning models and algorithms for big data classification* (pp.183 – 195) Springer International Publishing.

Suthaharan, S. (2016). *Machine learning models and algorithms for big data classification* pp.207-220). Springer International Publishing.

Suthaharan, S. (2016). *Machine learning models and algorithms for big data classification* (pp. 237 - 266) (pp.160-162) (pp.183 – 195)(pp.207-220). Springer International Publishing.

Rodriguez, J., Perez, A., & Lozano, J. (2010). *Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation*. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, *32*(3), 569-575. doi: 10.1109/tpami.2009.187

Wickham, H., & Grolemund, G. (2017). *R for data science*. Beijing . O'Reilly.

Witten, I. and Frank, E., 2017. *Data Mining*. San Francisco, Calif.: Morgan Kaufmann, pp.179 - 192.

APPENDIX

Classification Thomas Madeley.Rmd

ASDM Classification Code Thomas Madeley.pdf

Click above to open PDF to view code.

[Classification: Accurately Diagnosing Breast Cancers Using Machine Learning Models (shinyapps.io)]

Part 2: Association Rules Mining: Using Association Rules To Devise Marketing Strategies In a Hospitality Venue

Abstract

2020 has been an incredibly difficult year for the hospitality industry in the UK. As a non-essential part of life, it was one of the first industries to have restrictions placed upon it during the COVID-19 pandemic (Dunn, Allen, Cameron, Malhotra, and Alderwick, 2020). Therefore, it is more important than ever for businesses to devise intelligent marketing strategies to maximise revenue and stay competitive. This paper will assess whether sales data from a small business can be used to mine further insights and to discover insights that can be leveraged to create business strategies that can increase food revenue and improve customer experience using association rules mining.

Introduction

This paper explores a data set exported from a Manchester based bar and restaurant. This paper's question was derived from discussing the business needs with the company directors. They specified a clear business goal of increasing business revenues and specifically revenues from food products. The focus of the association rules mining analysis was therefore to mine insights that may guide their business and marketing strategies. The dialogue between a venue and their customers is conducted via marketing. Marketing can control how customers perceive a venue, how much they may spend at a venue or whether they will visit a business at all (Katsigris & Thomas, 2012). Marketing is also expensive, with hospitality businesses spending between 1% and 5% of revenue on marketing and promotions (Katsigris & Thomas, 2012). It is therefore important that a venues marketing is well informed and carefully targeted. Hospitality venues generate a lot of sales data through their point of sales (POS) systems. Typically, this data is only used for tracking stock levels and generating revenue reports. This paper will explore sales data from a hospitality venue and uses exploratory analysis and the Apriori algorithm to access whether sales data can be used to discover insights that can be used to build marketing strategies to increase food revenues and improve customer experience.

Association Rules Mining and Apriori

Association rules are relationships between sets of products, constructed from the combinations in which they have been purchased in the past. Association rules may be discovered from or 'mined' from transactional datasets. Association rules have two components, an antecedent {X} and a consequent {Y} (Witten & Frank, 2017). The strength and value of an association rule is measured with 3 key parameters: Confidence, Support (Sometimes knows as Coverage and Accuracy) and Lift (Witten & Frank, 2017). The confidence of a rule defines the probability that the consequent will be

purchased, given that the antecedent has already been purchased. For example, it would be expected that an association rule for {Bread} → {Milk} would have a high confidence as they are commonly purchased together. A rule for {Bread} → {Engine Oil} would be expected to have a low confidence. Support defines how frequently a particular combination of items are purchased together within the dataset. High support would be notable as it would indicate that a larger proportion of the total transaction contained the items. A low support may also be notable as it would indicate a lack of transactions containing the items (Han, Kamber & Pei, 2011 p245-246). For example, the rule below:

$$\{Bread\} \rightarrow \{Milk\} \ [Support = 2\% \ Confidence = 60\%]$$

Figure 24 - Example Association Rule

The above rule shows that 2% of the total transactions contained both bread and milk and 60% of customers who bought bread, also bought milk.

Lift is a measure of whether an item set is independent of one another. It can be defined as the probability of the items in the association rule being purchased together, adjusted for the popularity of the consequent in the dataset. Lift is generally the best measure of the strength of an association rule. A lift of 1 would indicate there is no positive or negative relationship and the antecedent and consequent are independent of each other. A lift greater than 1 would indicate that the consequent is positively correlated with the antecedent and is more likely to be purchased given that the antecedent has been purchased. A lift of less than 1 would indicate that the consequent is negatively correlated with the antecedent and is less likely to be purchased given that the antecedent has been purchased (Han, Kamber & Pei, 2011, p245-246). It should also be noted that a high lift rule is only as valuable as the support for that rule.

$$Association\ Rule = \{X\} \rightarrow \{Y\}$$

$$Confidence(\{X\} \rightarrow \{Y\} = \frac{N\ Transactions\ Containg\ X\ and\ Y}{N\ Transactions\ Containing\ X}$$

$$Expected\ Confidence(\{X\} \rightarrow \{Y\} = \frac{N\ Transactions\ Containing\ X}{Total\ Number\ of\ Transactions}$$

$$Support(\{X\} \rightarrow \{Y\} = \frac{N\ Transactions\ Containg\ X\ and\ Y}{Total\ Number\ of\ Transactions}$$

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{(N\ Transactions\ Containg\ X\ and\ Y)/(N\ Transactions\ Containing\ X)}{Fraction\ of\ Transactions\ Containing\ Y}$$

Lift may also be expressed as:

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{Confidence}{Expected\ Confidence}$$

Figure 25 - Relevant Equations Used

(Han, Kamber & Pei, 2011, p255-p256)

Apriori is the name of the algorithm used to mine association rules from a dataset. Apriori is 'naïve' algorithm meaning that it iterates over all items in the dataset one by one, this means it is a slow algorithm. Iterating over all the combinations of items in the dataset, support, confidence and lift are calculated and then output. These rules can be filtered or 'pruned' by a user by setting limits on minimum confidence and support required for a rule (Han, Kamber & Pei, 2011). This would create issues later in the analysis due to the large number of unique products in the dataset used: high number of dimensions over which the algorithm would need to iterate.

Dataset and Tools

The dataset used for this paper comes from a Manchester, UK based bar and restaurant venue called Dive Bar & Grill. Permission for use of the data for research purposes has been granted by the company directors. The dataset contains 23 features 20930 rows, or entries, each entry represents an individual product sale. The dataset is a complete export from the point-of-sale system (POS), known as 'Lightspeed Restaurant', used in the venue and is in comma separated value (CSV) format. The CSV export is an integrated function of the POS. Individual transactions can be identified by a common receipt ID number. The data covers all transactions between 22/03/2020 and 04/11/2020. These transactions include customer transactions and 'internal' transactions such as 'wastages' or discounted staff purchases products discarded due to a spillage, past the sell by date or staff meals. The features are a mixture of numeric and character formats. The dataset contains no customer information but does however contain the first names of staff members. To address privacy concerns this column will be removed during the analysis.

Feature Name	Format	Description
Company_ID	Numerical	Unique POS Company Identifier
Company_Name	Character	The trading name of the company
Receipt_ID	Numerical	Identifying number of each receipt or transaction
Created_By	Character	Staff name that created the receipt
Creation_Date	Character	Contains the date and time of receipt creation in character format
Modification_Date	Character	Contains the date and time of receipts last modificatoin in character format
Last_modified_by	Character	Staff name that last modified the receipt
Product_ID	Numerical	Unique ID number of each product
Status	Character	Status of a receipt, sent implying sent to the kitchen or bar to be fulfilled
Name	Character	Product Name
Kitchen_Name	Character	Product Name as it appears on the kitchen printer
Quantity	Numerical	Quantity ordered
Tax_Exclusive_Price	N/A	Unused column with missing values
Tax_Inclusive_Price	Numerical	Price of product including tax in pounds sterling
Total_Tax_Exclusive_Price	N/A	Unused column with missing values
Total_Tax_Inclusive_Price	Numerical	Price of product including tax in pounds sterling
Tax_Percentage	Numerical	Tax rate for the item in %
Category	Character	The product category to which the product belongs
Category_Type	Character	Revenue stream the product belongs to: Food or Drinks
Seat_Number	Numerical	Unused column with 0 values
Course_Number	Numerical	Numerical value representing which course the product was ordered for. 0 Drinks, 1 Starter, 2 Main Course, 3 Dessert
Extra	Character	Optional Extra comments written by the staff member
PLU	Character	Unique product code used as an ID code by the POS software

Figure 26 - Metadata

The key tools used in this paper are RStudio v1.3.1093 with R v4.0.3 and SAS Enterprise Miner Workstation v14.3. Within RStudio the key package used were Stringr v1.4.0, arules v1.6-6, arulesviz v1.3-3, dplyr v1.0.2, ggplot2 v3.3.2 and lubridate v1.7.9.2. SAS Enterprise Miner Workstation was used in its base form. The complete RStudio code used is available as an embedded PDF file in the appendix (See appendix).

Data Preparation and Feature Selection

All preparation steps were completed in RStudio. As this dataset has been retrieved directly from a live system, it required significant cleaning and modification to reach usable state. The dataset was imported into R using the 'read.csv' function and the dimensions and structure inspected using the 'dim' and 'str' functions. It was noted that there were several columns consisting of missing values or 0 values only. It was also noted that several of these columns such as ones relating to prices, seat number and taxes, were also not relevant for association rules mining. These columns were initially retained as it would allow for filtering and refining of the entries before analysis but were later dropped. Due to COVID-19, the business was unable to trade from 24/03/2020 to 23/09/2020. The business was able to reopen on 24/09/2020 and was subsequently forced to close on 1/11/2020. Due to the closures only transactions between the above dates could be considered as real sales for analysis, anything outside this date is an 'internal' transaction. While the dataset contained a Creation_Date column unfortunately, this was in a character format and included the creation time (DD/MM/YYYY HH:MM), meaning that it was unable to be used as a filter. Using the Stringr library, the character column was split on the space between date and time, saving date and time to separate columns. Then the date column was converted into a date using the lubridate package, and then reassigned. Once in correct date format (YYYY-MM-DD), the dataset was filtered for

transactions between 24/09/2020 and 1/11/2020.

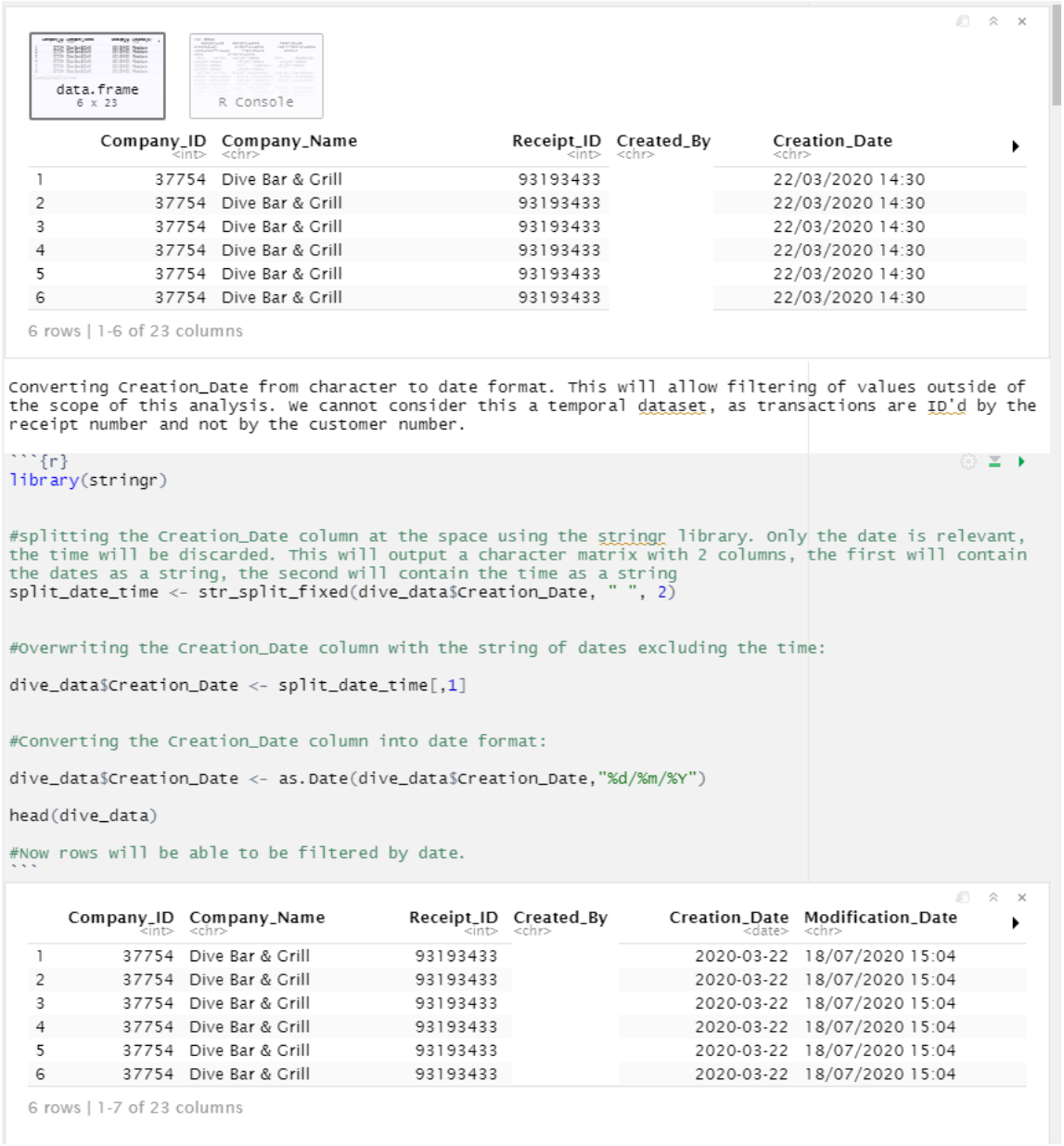


Figure 27 - Date Conversion Code Snippet

The ‘‘Category_Type’ ‘column was the next to be identified as needing filtering. The column contained entries with a ‘Category_Type’ of ‘Unknown’ and further entries with a ‘Category_Type’ of Discounts. Upon closer inspection of these entries, it was noted that the ‘Unknown’ products were invalid rows and were dropped. The products with the ‘Category_Type’ of Discounts were identified as not true sales. Consultation with the client showed that these ‘products’ were not products but in fact how the POS system records a discount applied to a transaction which could be an internal transaction like wastage. These Discount entries were filtered out and removed. Some entries were found to contain other items that were also not true sales. Further consultation showed that these entries were selection products, these entries are created when a user uses a shortcut selection button on the POS to quickly access a product. These selection products were filtered and removed

using Stringr library's 'str_detect' function one the 'Name' column. 'str_detect' was used as it allows for finding partial matches of strings to be found. Further entries were removed as they contained a sale price less than £0, which is likely to have been an error or a discounted product missed by the previous filter.

Once all the remaining entries were established to be valid for the timeframe and genuine customer sales, columns irrelevant to Apriori analysis could be dropped: (Company_ID, Company_Name, Created_By, Modification_Date, Last_modified_by, Product_ID, Status,Kitchen_Name, Tax_Exclusive_Price, Total_Tax_Exclusive_Price, Total_Tax_Inclusive_Price, Tax_Percentage, Seat_Number, Extra", PLU). The remaining data frame contained 8 columns and 13668 entries; the data frame was written to a new CSV for future use in SAS. The most critical remaining columns were 'Receipt_ID' containing the transaction number, 'Name' containing the product name and 'Product_Category' containing the product category.

The dataset was temporarily divided into food and drink product sales, number of sales for each product and each product category was then plotted using 'dplyr' library's 'group_by' and 'summarise' functions and the 'ggplot2' library so that the most and least sold products could be identified for both food and drinks products.

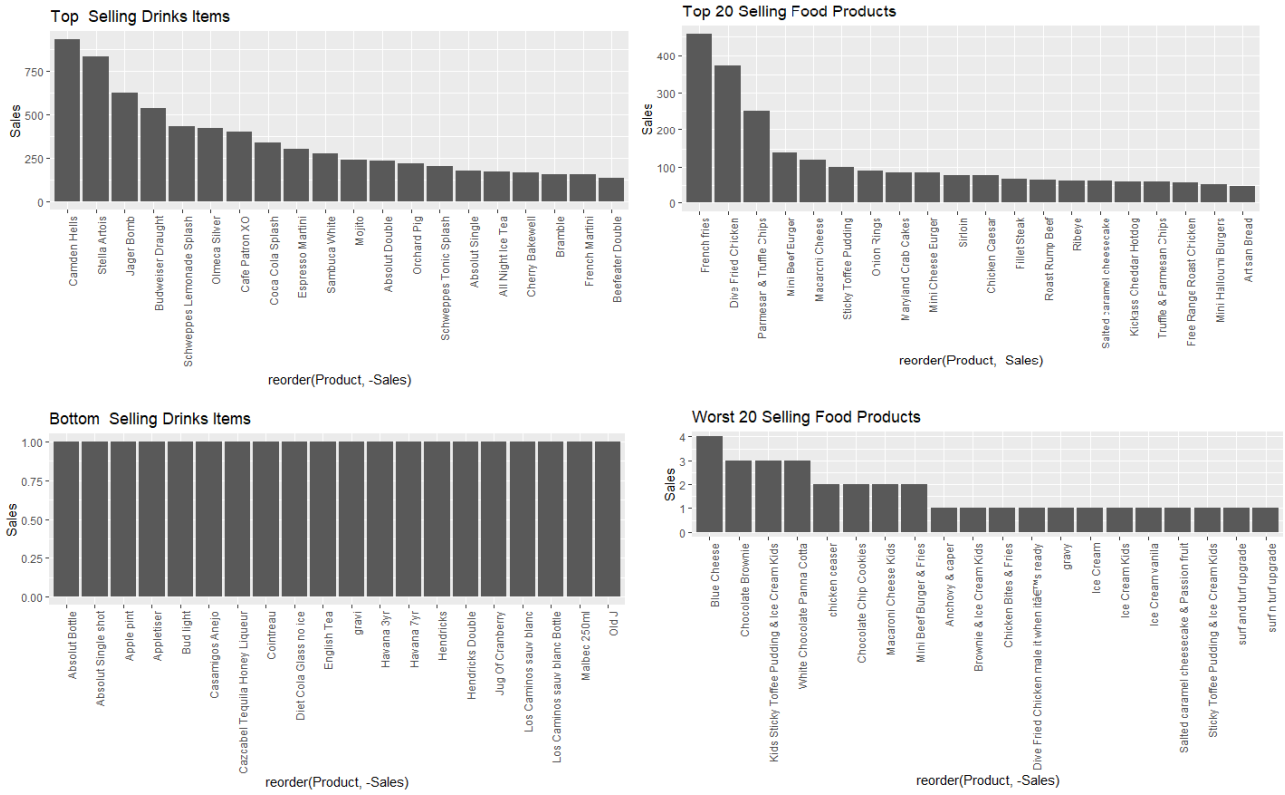


Figure 28 - Sales Distributions

It was noted that due to the large number of individual products, 291, there was a very large range of units of each product sold. While the large number of products would enable a lot of rules to be generated, these rules may be too specific to be of use. Additionally, support for said rules would be low, as there were many possible item sets. It was decided to create a product categories dataset to analyse in parallel. The distributions for sales by category were further plotted using 'ggplot2'.

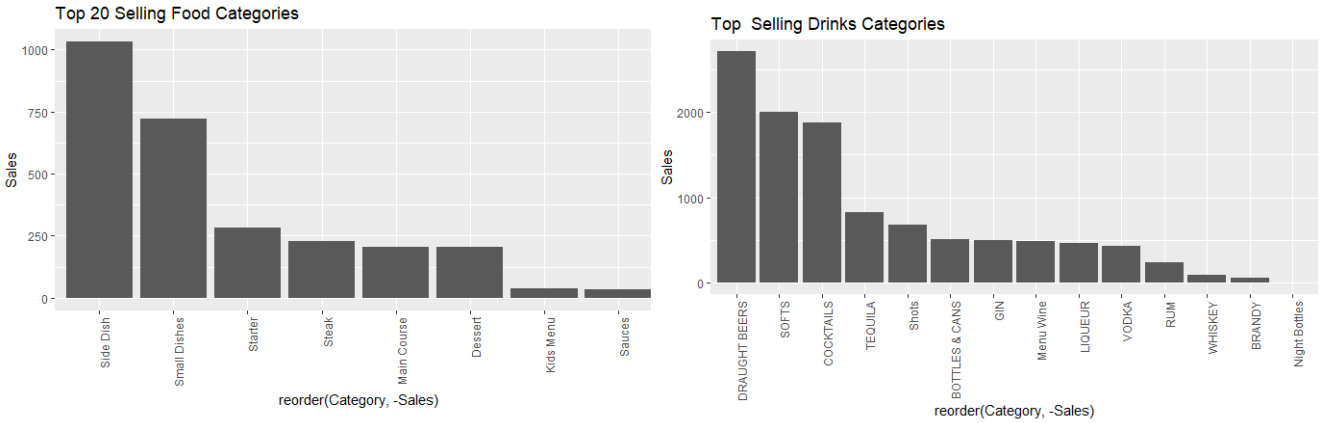


Figure 29 - Category Sales Distributions

The range of 291 product's sales have now been condensed into 8 food categories and 14 drinks categories which may allow for more broad association rules to be mined.

The final data pre-processing step was to convert the remaining data frames into their final formats to be used in the Apriori algorithm. To do this the data frames would need to be pivoted from 'long' format into 'wide' format. The data frame is 'long' because the transactions are currently split over multiple entries: each product sold is listed in a new entry with a column indicating a 'Receipt_ID' identifying which transaction it belongs to. The 'Receipt_ID' and 'Name' columns were selected from the remaining data using 'dplyr' library's 'select' function. This data was the pivoted using the 'table' function. This had the effect of grouping each transaction onto a single row with a column representing each product, within each product column there is a numerical value representing the number of each product purchased. The Apriori algorithm requires these continuous columns to be converted into binary values in order to create association rules. The 'apply' function in combination with the 'as.logical' function to convert these continuous into a logical 'TRUE' or 'FALSE' value. True if the numerical value within the column was greater than 0 representing a purchase, else 0. This process was repeated to create a data frame with using 'Receipt_ID' and 'Category' to be analysed in parallel. This successfully pivoted the data from 'long' format (2 Columns, 13668 Rows) into wide format with logical values in the columns (291 Columns, 1640 Rows (transactions) for the individual product data frame and 23 columns and 1640 rows for the category data frame). The product and categorical data frames were then carried forward for analysis, the head of each data frame before and after transformation is shown below.

Long Format

	Receipt_ID <int>	Name <chr>
1	93193433	Prosecco Bottle
2	93193433	Budweiser Draught
3	93193433	Budweiser Draught
4	93193433	Budweiser Draught
5	93193433	Budweiser Draught
6	93193433	Camden Hells

6 rows

Wide Format with Logical Columns

	Absolut Bottle <lgl>	Absolut Double <lgl>	Absolut Double no ice <lgl>	Absolut Kurant <lgl>
1	FALSE	FALSE	FALSE	FALSE
2	FALSE	FALSE	FALSE	FALSE
3	FALSE	TRUE	FALSE	FALSE
4	FALSE	FALSE	FALSE	FALSE
5	FALSE	FALSE	FALSE	FALSE
6	FALSE	FALSE	FALSE	FALSE

6 rows | 1-5 of 291 columns

Figure 30- Long to Wide Individual Products

Long Format

	Receipt_ID <int>	Category <chr>
1	93193433	Menu Wine
2	93193433	DRAUGHT BEERS
3	93193433	DRAUGHT BEERS
4	93193433	DRAUGHT BEERS
5	93193433	DRAUGHT BEERS
6	93193433	DRAUGHT BEERS

6 rows

Wide Format With Logical Columns

	BOTTLES & CANS <lgl>	BRANDY <lgl>	COCKTAILS <lgl>	Dessert <lgl>	DRAUGHT BEERS <lgl>	GIN <lgl>
1	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
2	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
3	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
5	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
6	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE

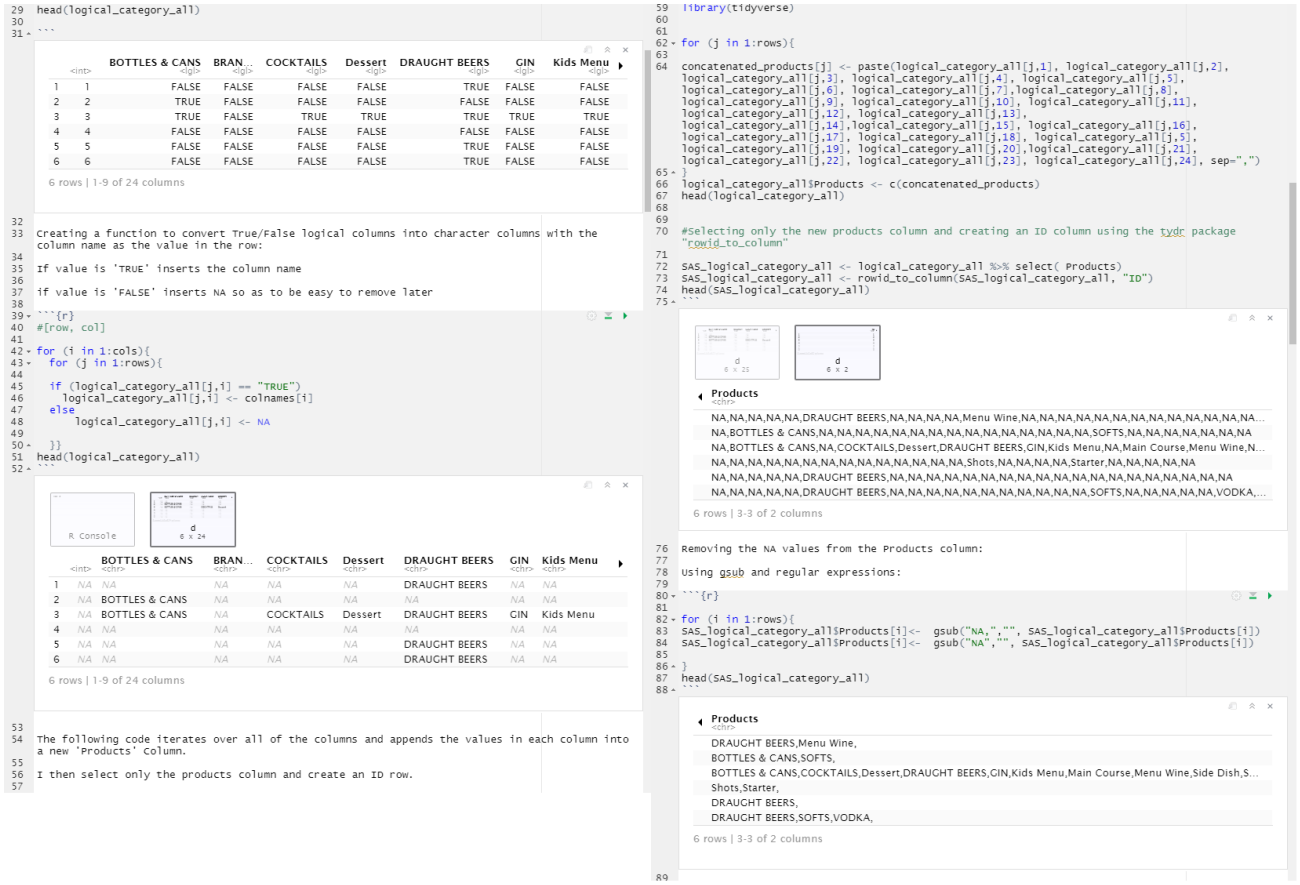
6 rows | 1-8 of 23 columns

Figure 31 - Long to Wide Categories

Data Preparation for SAS

While the above logical, wide format is suitable for Apriori implementation in R, SAS requires a very different format for implementation. The SAS 'Association Rule' node requires the data in 'long' format. Although the original dataset was in long format, many pre-processing and filtering steps had already been completed in R. Rather than using the original dataset and repeating preparation steps, it was decided that the already pre-processed datasets for products and categories would be mutated for use in SAS.

The product and categorical logical data frames created for R analysis were written to a csv using the 'write.csv' function. The data frames were imported into a separate script and a series of functions were written to covert the data frames into the appropriate SAS format. The first function iterated over all the rows and all columns, using and if / else statement to replace any instances of the logical 'TRUE': a product sale, with the column name in character format. The function also replaced any instance of 'FALSE': not a sale, with NA (a missing value). The next function concatenated all of the new character columns into a single 'Product' column. The 'Product' column now consisted of a list of the NA values and column names separated with commas. These NA values were then removed using the 'gsub' function. Finally using the 'splitstackshape' library's 'cSplit' function, the 'Product' column lists were split into long format: One product per row with a common 'ID' representing the transaction number. This process was repeated to create data frames for products, categories, food categories and drinks categories. These data frames were then written to csv for later use in SAS. The code used to complete this is available in the appendix.



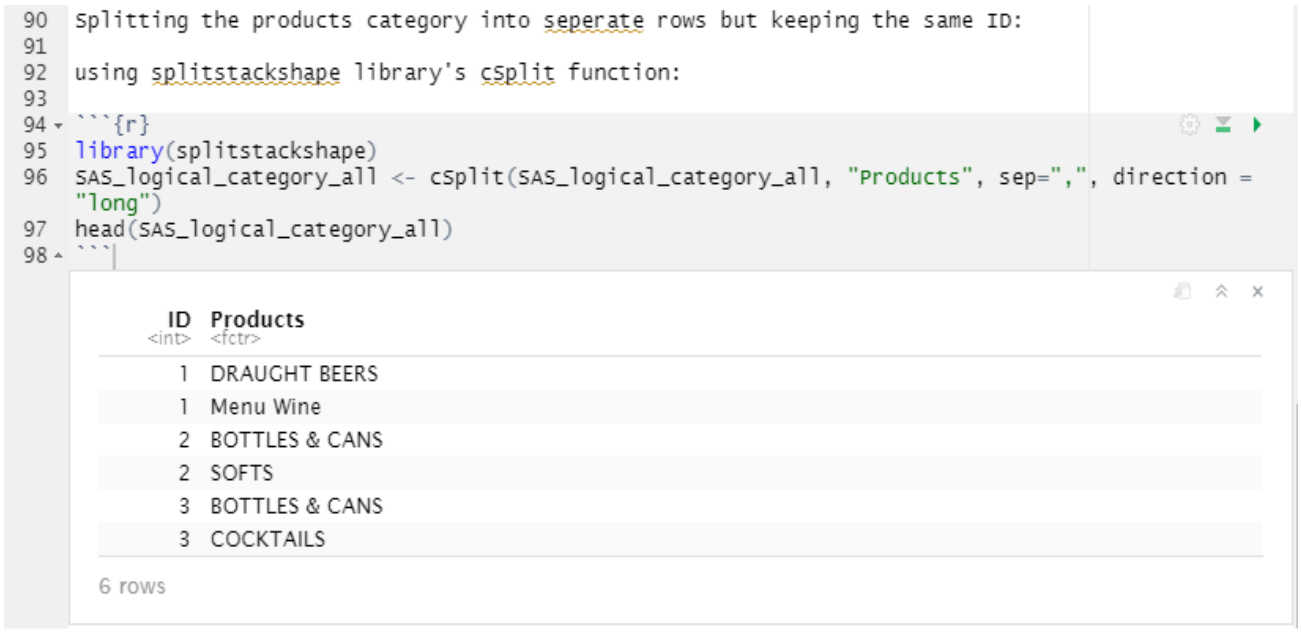


Figure 32 - SAS Data Pre-Processing

Implementation in R

Once the extensive data preparation was completed the product and category data frames were both carried forward for analysis in R. The Apriori algorithm was implemented for both datasets using the 'arules' packages's 'apriori' function.

The products data frame was analysed first, using the 'apriori' with no hyperparameters altered. This returned 0 rules. Upon inspecting the 'apriori' function, it was noted that the default minimum support and confidence are 0.1 and 0.8, respectively. Due to the data frame containing many unique products (291), it would be expected that a minimum support of 0.1 would not return many rules as an antecedent and consequent would need to appear in 10% of transactions in order to generate a rule. Despite this, the data set should be able to generate rules with high confidence as confidence is only relative to the number of transactions containing the antecedent. In order to generate more rules, the minimum support was lowered to 0.01 and the minimum confidence was lowered 0.6. In addition, the minimum and maximum length of the rules were set to 2 and 5, respectively. This returned 20 rules. The rules were sorted using the 'arules' library's 'sort' function. Three rules objects were saved, containing the rules sorted by the three key parameters of support, confidence and lift in descending order.

The above procedure was repeated for the category data frame. The default parameters of support and confidence for the 'apriori' function returned 2 rules which is consistent with the reduction of columns: from 291 products to 23 categories. Carrying forward the hyperparameters from the products analysis resulted in 226 rules being generated. In order to refine the results, the support and confidence parameters were tuned to 0.01 and 0.6 respectively. Minimum length and maximum length were kept at 2 and 5 respectively.

In order to visualise the results, the 'arulesviz' package was utilised. The 'ruleExplorer' function creates an interactive R 'shiny' application. The 'ruleExplorer' function was used to generate plots and matrices in order to evaluate and access the rules generated. A 'ruleExplorer' application was created for both rule sets generated for the products and categories data frames.

Results in R

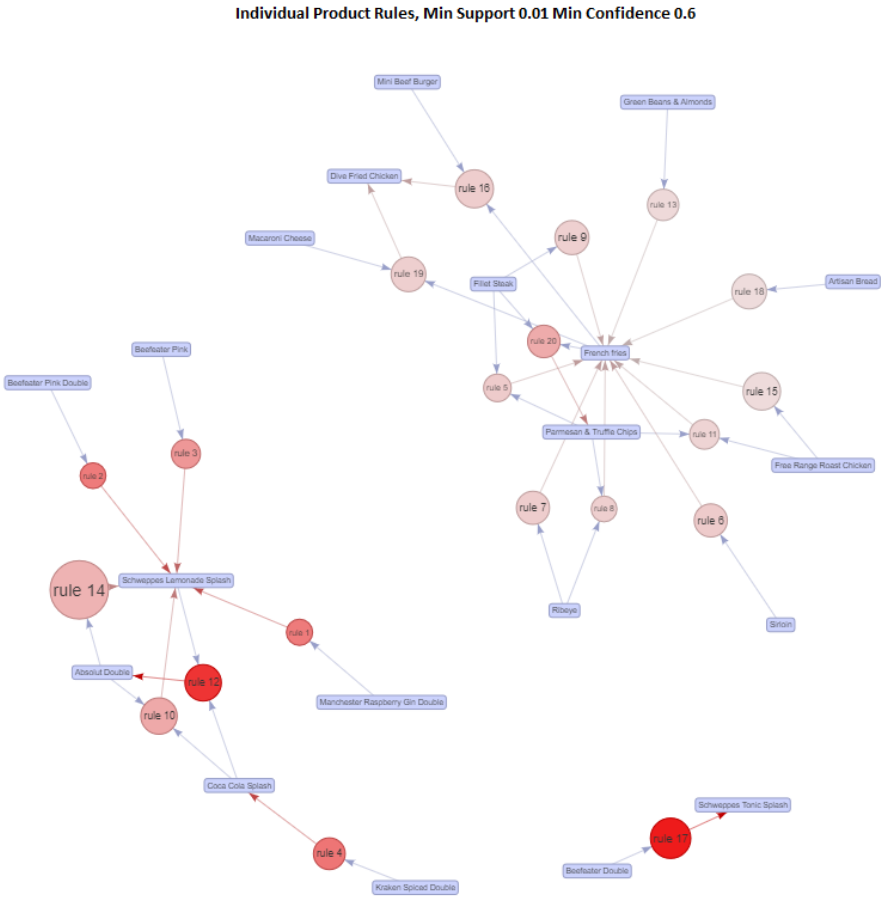
Initially the results from the products data frame were investigated. The 'apriori' function was implemented with a minimum support of 0.01 and a minimum confidence of 0.6 and generated 20 rules. The rules object generated by the 'apriori' function containing the association rules was sorted by lift and the rules were inspected.

Individual Product Rules Sorted by Lift Descending

	lhs		rhs	support	confidence	coverage	lift	count
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
[1]	{Beefeater Double}	=>	{Schweppes Tonic Splash}	0.02134146	0.6140351	0.03475610	10.828146	35
[2]	{Coca Cola Splash,Schweppes Lemonade Splash}	=>	{Absolut Double}	0.01707317	0.6511628	0.02621951	10.074594	28
[3]	{Kraken Spiced Double}	=>	{Coca Cola Splash}	0.01829268	0.7894737	0.02317073	8.142999	30
[4]	{Manchester Raspberry Gin Double}	=>	{Schweppes Lemonade Splash}	0.01097561	0.9000000	0.01219512	7.978378	18
[5]	{Beefeater Pink Double}	=>	{Schweppes Lemonade Splash}	0.01036585	0.8947368	0.01158537	7.931721	17
[6]	{Beefeater Pink}	=>	{Schweppes Lemonade Splash}	0.01463415	0.8000000	0.01829268	7.091892	24
[7]	{Absolut Double,Coca Cola Splash}	=>	{Schweppes Lemonade Splash}	0.01707317	0.7000000	0.02439024	6.205405	28
[8]	{Fillet Steak,French fries}	=>	{Parmesan & Truffle Chips}	0.01280488	0.6000000	0.02134146	6.074074	21
[9]	{Absolut Double}	=>	{Schweppes Lemonade Splash}	0.04085366	0.6320755	0.06463415	5.603264	67
[10]	{Fillet Steak,Parmesan & Truffle Chips}	=>	{French fries}	0.01280488	0.7777778	0.01646341	4.309309	21
[11]	{Sirloin}	=>	{French fries}	0.02012195	0.7674419	0.02621951	4.252043	33
[12]	{French fries,Mini Beef Burger}	=>	{Dive Fried Chicken}	0.01890244	0.6200000	0.03048780	4.167213	31
[13]	{Ribeye}	=>	{French fries}	0.02012195	0.7500000	0.02682927	4.155405	33
[14]	{French fries,Macaroni Cheese}	=>	{Dive Fried Chicken}	0.01524390	0.6097561	0.02500000	4.098361	25
[15]	{Parmesan & Truffle Chips,Ribeye}	=>	{French fries}	0.01036585	0.7391304	0.01402439	4.095182	17
[16]	{Fillet Steak}	=>	{French fries}	0.02134146	0.7291667	0.02926829	4.039977	35
[17]	{Free Range Roast Chicken,Parmesan & Truffle Chips}	=>	{French fries}	0.01036585	0.6800000	0.01524390	3.767568	17
[18]	{Green Beans & Almonds}	=>	{French fries}	0.01158537	0.6333333	0.01829268	3.509009	19
[19]	{Free Range Roast Chicken}	=>	{French fries}	0.01829268	0.6250000	0.02926829	3.462838	30
[20]	{Artisan Bread}	=>	{French fries}	0.01524390	0.6097561	0.02500000	3.378378	25

Figure 33 - Apriori Rules

It was noted that there are strong association rules between spirits and soft drinks: Beefeater Gin and tonic water. It was also noted that food items were commonly bought with french fries: Sirloin and French fries. While the rules generated have very high lift meaning they are of high predictive value on future sales, they simply confirm what could have been deduced with general intuition. What was most surprising is that there were no rules containing both a drink product and a food product, this indicates that there is little correlation between the food and drink purchases. Using the 'ruleExplorer' function the rules were plotted. The size of each node represents the support and the colour represents the lift, high lift is more red.



It can be seen that there are two distinct clusters of rules around ‘Schweppes Lemonade’ and ‘ French Fries’.

The products data frame was split into two separate data frames, one containing the drinks products and another containing the food products. These data frames were analysed independently. Using the same parameters for the ‘apriori’ function used in the complete database, the rules yielded were identical.

It was decided that a different direction would need to be taken for the analysis. As outlined in the data preparation section, the products were grouped into their respective sales categories. Using the ‘apriori’ function with the same parameters produced 244 rules. Condensing the products into their respective categories had the effect of decreasing the resolution of the analysis: decreasing the number of unique antecedents and consequents. In order to increase food sales, association rules containing both drink and food sales were first considered. These can be used to construct strategies that appeal to the existing dining customer base.

Category Rules Sorted by Lift Descending

	lhs <chr>	<chr>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
[1]	{COCKTAILS,Main Course,SOFTS}	=>	{Dessert}	0.01096892	0.7500000	0.01462523	12.185644	18
[2]	{Dessert,Menu Wine,SOFTS}	=>	{Main Course}	0.01096892	0.7826087	0.01401584	11.782210	18
[3]	{COCKTAILS,Dessert,Starter}	=>	{Main Course}	0.01035954	0.7727273	0.01340646	11.633445	17
[4]	{COCKTAILS,Main Course,Starter}	=>	{Dessert}	0.01035954	0.7083333	0.01462523	11.508663	17
[5]	{COCKTAILS,Main Course}	=>	{Dessert}	0.01706277	0.6829268	0.02498477	11.095871	28
[6]	{Dessert,SOFTS,Starter}	=>	{Main Course}	0.01340646	0.7333333	0.01828154	11.040367	22
[7]	{Main Course,SOFTS,Starter}	=>	{Dessert}	0.01340646	0.6666667	0.02010969	10.831683	22
[8]	{Main Course,Menu Wine,SOFTS}	=>	{Dessert}	0.01096892	0.6666667	0.01645338	10.831683	18
[9]	{COCKTAILS,Main Course,Side Dish}	=>	{Dessert}	0.01340646	0.6470588	0.02071907	10.513104	22
[10]	{Main Course,SOFTS}	=>	{Dessert}	0.02132846	0.6363636	0.03351615	10.339334	35
[11]	{Dessert,Side Dish,Starter}	=>	{Main Course}	0.01462523	0.6857143	0.02132846	10.323460	24
[12]	{Dessert,Menu Wine,Side Dish}	=>	{Steak}	0.01157831	0.7037037	0.01645338	10.219272	19
[13]	{Dessert,Starter}	=>	{Main Course}	0.01889092	0.6739130	0.02803169	10.145792	31
[14]	{Dessert,Side Dish,Starter}	=>	{Steak}	0.01462523	0.6857143	0.02132846	9.958028	24
[15]	{Dessert,Side Dish,SOFTS}	=>	{Main Course}	0.01462523	0.6486486	0.02254723	9.765435	24
[16]	{Main Course,Menu Wine}	=>	{Dessert}	0.01279707	0.6000000	0.02132846	9.748515	21
[17]	{Dessert,DRAUGHT BEERS}	=>	{Main Course}	0.01035954	0.6296296	0.01645338	9.479103	17
[18]	{COCKTAILS,Dessert,Side Dish}	=>	{Main Course}	0.01340646	0.6285714	0.02132846	9.463172	22
[19]	{Dessert,Side Dish,SOFTS}	=>	{Steak}	0.01462523	0.6486486	0.02254723	9.419756	24
[20]	{Dessert,Menu Wine}	=>	{Main Course}	0.01279707	0.6176471	0.02071907	9.298705	21

Figure 34 – Food Category Rules

Once again, the majority of rules had a consequent containing one of the most popular items: ‘Side Dish’ and ‘SOFTS’), following a similar patten to the analysis at the product level. It was noted that there is a strong relationship between an order containing cocktails and an order containing dessert. Using the ‘ruleExplorer’ function the rules containing Cocktails in the antecedent and Dessert as the consequent were plotted, minimum lift was set to 8. Similarly, there are strong links between orders containing wine and orders containing dessert. The size of each node represents the support and the colour represents the lift, high lift is more red.

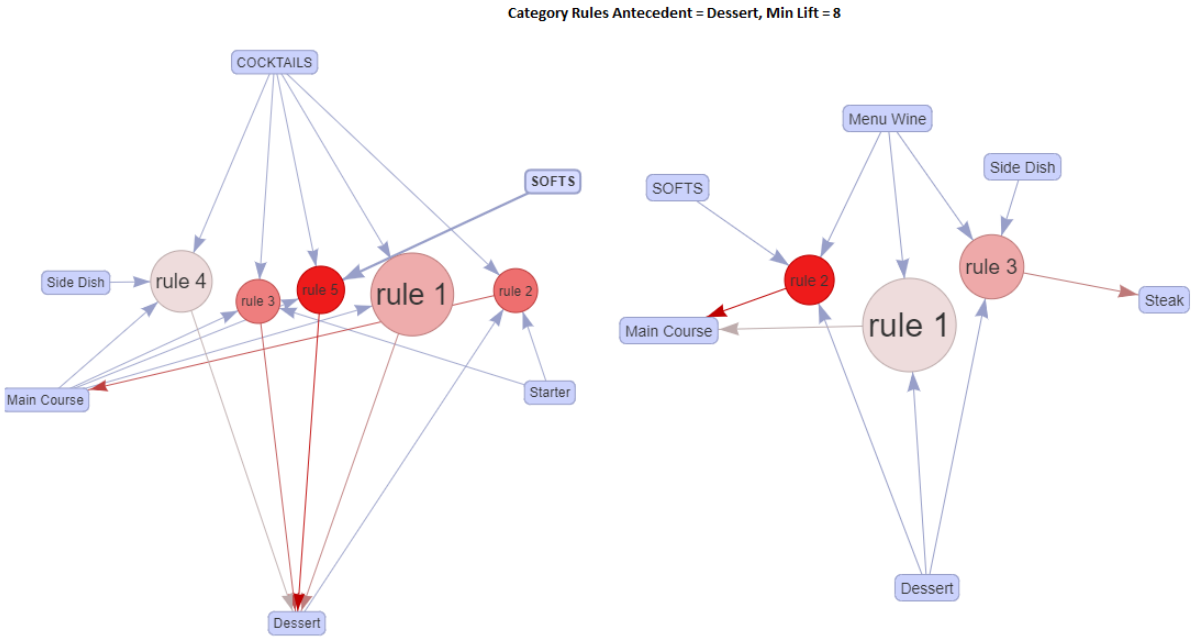


Figure 35 - Category Rules Containing Dessert

The customers ordering cocktails or wine and desserts, were also ordering either a main course or steak also. These could be classed as high spending customers as all the above categories are generally more expensive. Using this insight, a potential way to increase food sales would be to offer discounts on cocktails and wine when purchased in conjunction with a main meal or steak, which in turn is highly likely to result in the purchase of a dessert or starter or side dish also. Cocktails are the third most sold drinks product, and it would be expected that there would be some association with food sales just by sheer volume of sales. Wine is a low volume product with approximately one quarter of the sales volume when compared to cocktails and these rules could be considered more interesting. This will effectively present a value driven offer to encourage the group of customers who the business is already attracting to return more frequently.

Next it was decided to look for a lack of association between products which can be used to convert non-dining customers into dining customers. Draught beers were investigated first as this is the highest volume sales product. Using the 'ruleExplorer' function again the were filtered to show only rules containing 'DRAUGHT BEERS' in the antecedent. It was noted that only one rule containing draught beers had a lift of over 8. The minimum lift was reduced to 2.8 to generate more rules.

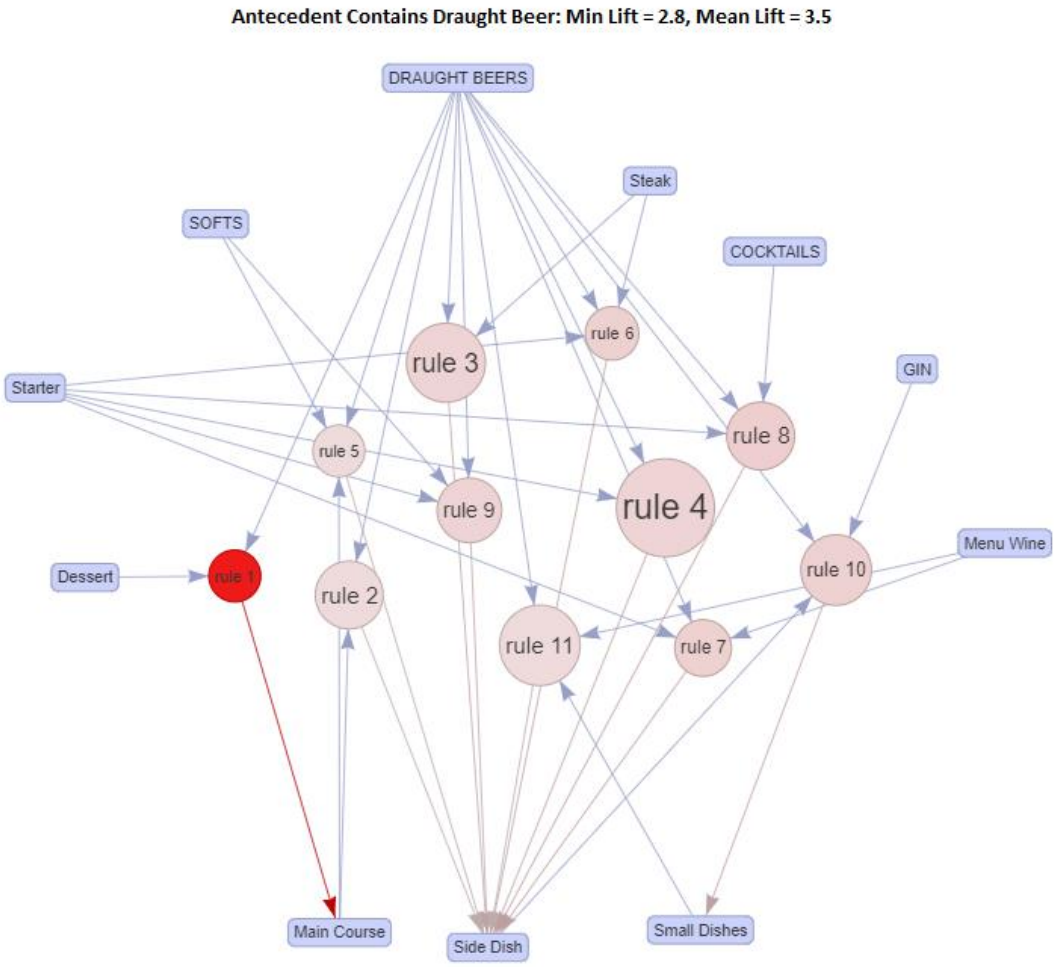


Figure 36 - Draught Beer Association Rules

While one rule with a very high lift shows association between draught beer and main course and dessert, most rules contain 'Side Dish' as the consequent. This suggests that customers ordering draught beer generally only order a small amount of food or 'snacks'. This suggest that draught beer drinks are typically not interested in the main meal food products offered. A potential strategy devised from this insight could be to alter the menu so as to offer items that are 'in between' a main course and a side dish in size or price, tempting into ordering a larger meal and spending more. The design of the menu could be reconfigured so that the 'Main Course' section would be highlighted to the customer, reducing the prevalence of the 'Side Dish' section as customers are already likely to order a side dish. Pricing could be used to incentivise purchasing of main meals with draught beers using discounting. As draught beers are the most frequently sold product, this could increase revenue even if there was a low conversion rate.

SAS Implementation

Due to the lack of insight generated at the individual product level, it was decided that a different approach would be taken with the SAS implementation. The SAS analysis would analyse the data set

at the category level, but instead of also analysing at the product level, the food and drinks categories would be split and analysed separately.

Using three file import nodes, the csv files previously created in R described in the 'SAS Data Preparation' section were imported for 'All Categories', 'Drinks Categories' and 'Food Categories'. For each of the input nodes, the variable roles were altered. The 'ID' column contains the transaction number and was set to the ID role. The 'Products' column contains the product categories purchased in a given transaction, this was set to the 'Target' role. An additional column named 'VAR1' was recognised by SAS. Using the 'File Import' (FI) node's 'Exported Data' function to view the data frame, it was evident that this 'VAR1' column contained a numerical index of each row which was unintentionally included when the CSV files were written from R. As this column was not needed, the variable role for it was set to 'Rejected': it will not be used in analysis. In the 'Score' section of the FI node, the role of the data was set to 'Transaction' as this is transaction data. This was repeated for all three of the CSV files.

The FI nodes were then run and the 'Results' page for each of them viewed. Using the 'Report' function a 'Class Targets' plot was generated in order to inspect the frequency of each category. 'Response' was changed from percentage to frequency count.

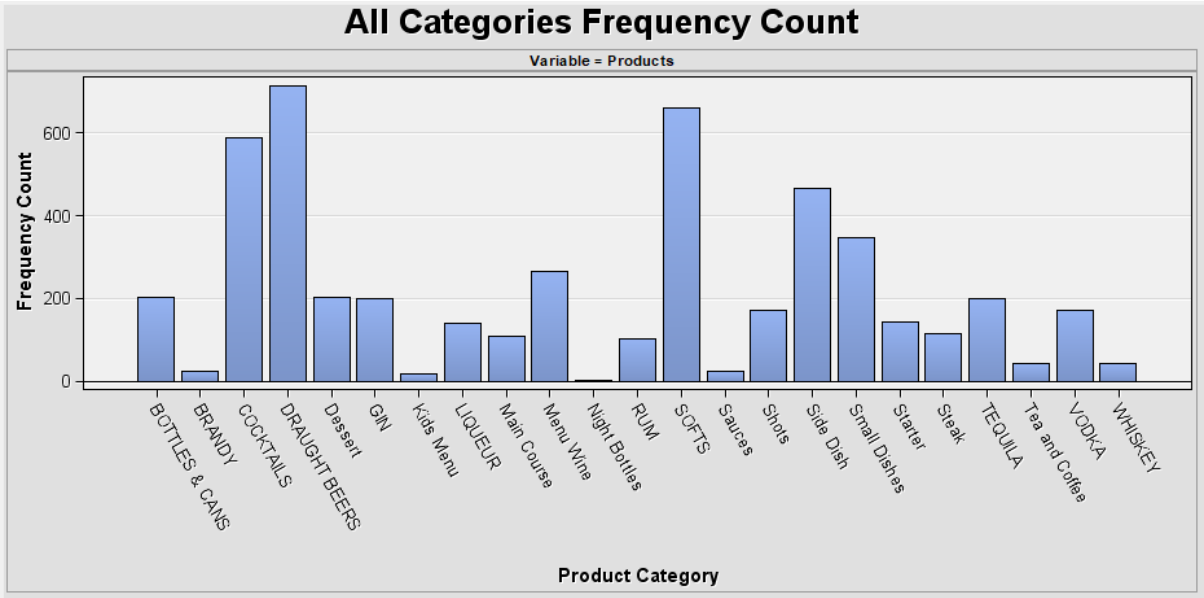


Figure 37 - SAS Sales Distribution

It was noted that the certain drinks categories such as 'COCKTAILS' and 'DRAUGHT BEERS' have a much higher frequency that other food or drink product categories.

A pair of 'Association' nodes was attached to each of the FI nodes. The first of which was used in its default state and the rules generated inspected. The second node was used to fine tune the rules returned. In the 'Association' section of the node, the 'Maximum Items', 'Minimum Confidence Level', and 'Support Percentage' were tuned. 'Minimum Confidence Level' was set to 60% for all of the data frames as it was decided that rules below this confidence level would not be considered. 'Support Percentage' was tuned to 2%. This created many rules for each of the data frames which could later be filtered and accessed.

Results in SAS

Using the 'Results' tab of the 'Association' node the Drinks Categories data was first analysed. It was noted that there were several rules with 100% confidence indicating a perfect correlation: Every time the antecedent was purchased, the consequent was also purchased. Unfortunately, many of the rules contained 'SOFTS' and the consequent and were deemed of little value. The 'Link Graph' was plotted, the size of the nodes corresponds to the number of transactions and the thickness of the linking lines corresponds to the confidence.

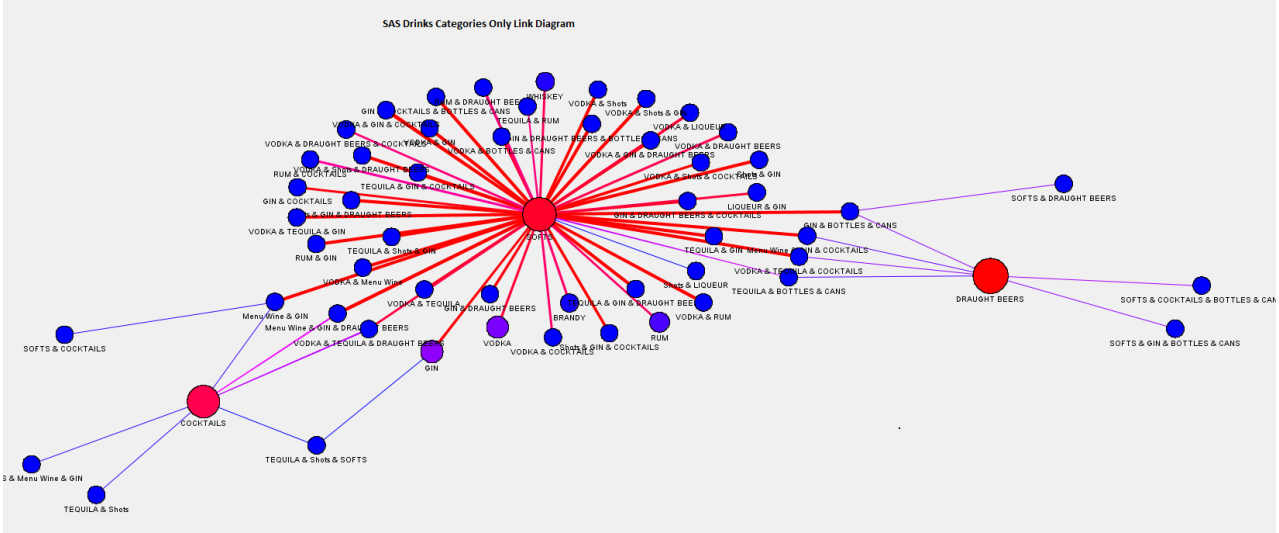


Figure 38- SAS Drinks Association Rules

The analysis of the Drinks Categories data was halted at this point as it was deemed not useful in answering this papers question.

Repeating the process with the Food Categories data some more relevant and unexpected rules were generated. It was noted that, when sorted by lift, all the top ten rules contained the 'Kids Menu' category.

Relations	Expected Confidence(%)	Confidence(%)	Support(%)	Lift ▼	Transaction Count	Rule
4	8.42	100.00	1.53	11.87	10.00	Starter & Kids Menu ==> Main Course & Dessert
4	7.04	76.92	1.53	10.92	10.00	Main Course & Kids Menu ==> Starter & Dessert
4	8.42	76.92	1.53	9.13	10.00	Kids Menu & Dessert ==> Starter & Main Course
3	8.42	66.67	1.84	7.92	12.00	Kids Menu ==> Main Course & Dessert
3	15.47	100.00	1.53	6.47	10.00	Starter & Kids Menu ==> Dessert
4	15.47	100.00	1.53	6.47	10.00	Starter & Main Course & Kids Menu ==> Dessert
3	16.69	100.00	1.53	5.99	10.00	Starter & Kids Menu ==> Main Course
4	16.69	100.00	1.53	5.99	10.00	Starter & Kids Menu & Dessert ==> Main Course
3	15.47	92.31	1.84	5.97	12.00	Main Course & Kids Menu ==> Dessert
3	16.69	92.31	1.84	5.53	12.00	Kids Menu & Dessert ==> Main Course
2	15.47	72.22	1.99	4.67	13.00	Kids Menu ==> Dessert
2	16.69	72.22	1.99	4.33	13.00	Kids Menu ==> Main Course
4	16.69	68.57	3.68	4.11	24.00	Starter & Side Dish & Dessert ==> Main Course
3	17.30	70.59	1.84	4.08	12.00	Side Dish & Sauces ==> Steak

Figure 39 - SAS Food Rules Sorted by Rules

These rules imply customers bringing children to the venue generally order a large amount of food. The transaction count, and therefore support, was low at around 1.5%. These rules could be utilised to increase food sales by marketing the venue as more child friendly and family focused. Therefore, bringing in more customers with children with families who are likely to purchase food items. It was also noted that many rules were centred around the 'Side Dish' category, which was to be expected as it is the most sold product.

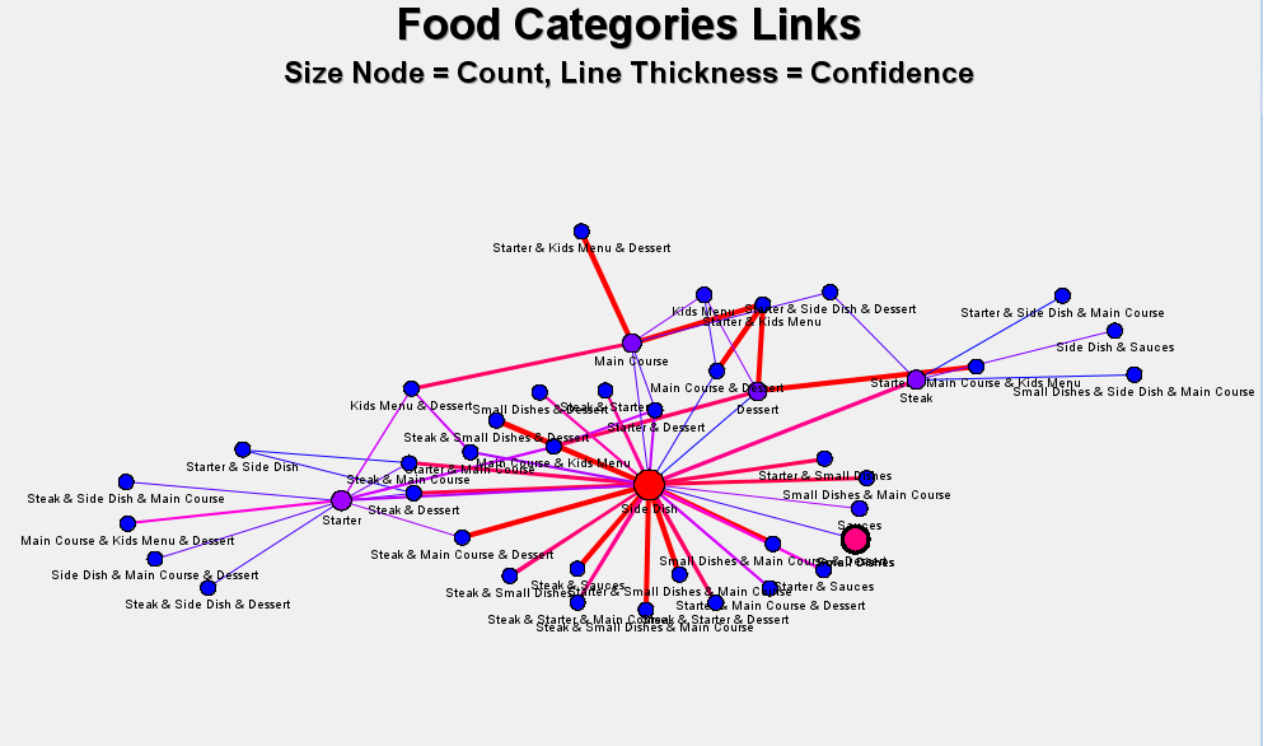


Figure 40 - Food Category Association Rules

Finally the rules generated for the data frame containing both food and drinks categories was analysed. Using the 'Results' function of the 'Association' node, the table of rules was inspected. To

gain an understanding of the items most purchased together, the table of rules was sorted by support.

Relations	Expected Confidence(%)	Confidence(%)	Support(%) ▼	Lift	Transaction Count	Rule
2	28.34	75.00	6.58	2.65	108.00	Starter ==> Side Dish
2	28.34	89.38	6.15	3.15	101.00	Steak ==> Side Dish
3	28.34	90.28	3.96	3.19	65.00	Starter & SOFTS ==> Side Dish
3	39.98	100.00	3.35	2.50	55.00	Side Dish & GIN ==> SOFTS
3	28.34	94.64	3.23	3.34	53.00	Starter & COCKTAILS ==> Side Dish
3	28.34	88.33	3.23	3.12	53.00	Steak & Starter ==> Side Dish
3	28.34	98.00	2.99	3.46	49.00	Steak & SOFTS ==> Side Dish
4	28.34	72.06	2.99	2.54	49.00	Small Dishes & SOFTS & DRAUGHT BEERS ==> Side Dish
3	39.98	100.00	2.99	2.50	49.00	VODKA & GIN ==> SOFTS
4	28.34	76.19	2.93	2.69	48.00	Small Dishes & SOFTS & COCKTAILS ==> Side Dish
3	39.98	100.00	2.86	2.50	47.00	TEQUILA & GIN ==> SOFTS
3	28.34	90.20	2.80	3.18	46.00	Starter & DRAUGHT BEERS ==> Side Dish
3	28.34	91.67	2.68	3.23	44.00	Steak & COCKTAILS ==> Side Dish
3	39.98	100.00	2.68	2.50	44.00	Small Dishes & GIN ==> SOFTS
3	28.34	78.18	2.62	2.76	43.00	SOFTS & Main Course ==> Side Dish
3	28.34	74.55	2.50	2.63	41.00	Starter & Main Course ==> Side Dish
3	39.98	100.00	2.50	2.50	41.00	VODKA & Side Dish ==> SOFTS
3	28.34	97.50	2.38	3.44	39.00	Starter & Menu Wine ==> Side Dish

Figure 41 - Combined Categories Association Rules by Support

It was noted that the highest support rules contained either 'Side Dish' or 'SOFTS' as the consequent, this correlates with the distribution of sales as these are the most sold products. As side dishes are very likely to be purchased with other food items, a potential method of increasing food sales would to be to incentivise the purchase of additional side dishes through pricing.

The rules for the all categories data frame were then plotted in 3 dimensions, with X,Y and Z representing support, confidence and lift respectively.

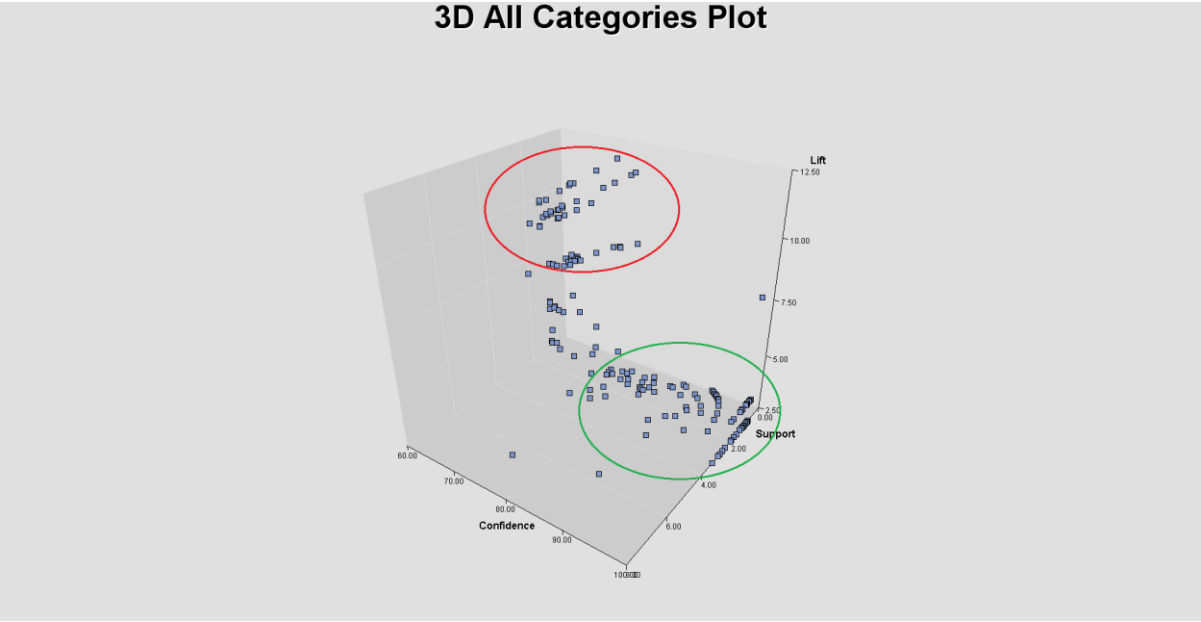


Figure 42 - Combined Categories 3D Plot

There were 2 distinct clusters of rules identified. The first cluster, in red, consists of rules with a high lift of over 7.5 despite the rules having a reasonably low confidence of between 60% and 75%. This implies the expected confidence of these rules is low, meaning sales volumes of the items concerned are relatively low. The second cluster, in green, contains rules of a much lower lift of between 2.5 and 3. The confidence of these rules is generally higher than the first cluster, with some even approaching 100% confidence. 100% confidence would imply that every time the antecedent was purchased, the consequent was also purchased. That these rules have a higher confidence, but a lower lift implies they have higher expected confidence: they are more commonly purchased products.

Relations	Expected Confidence(%)	Confidence(%)	Support(%)	Lift ▼	Transaction Count	Rule
4	6.15	75.00	1.10	12.19	18.00	SOFTS & Main Course & COCKTAILS ==> Dessert
4	6.64	78.26	1.10	11.78	18.00	SOFTS & Menu Wine & Dessert ==> Main Course
4	6.64	77.27	1.04	11.63	17.00	Starter & Dessert & COCKTAILS ==> Main Course
4	6.15	70.83	1.04	11.51	17.00	Starter & Main Course & COCKTAILS ==> Dessert
3	6.15	68.29	1.71	11.10	28.00	Main Course & COCKTAILS ==> Dessert
4	6.64	73.68	0.85	11.09	14.00	Steak & SOFTS & Menu Wine ==> Main Course
4	6.64	73.33	1.34	11.04	22.00	Starter & SOFTS & Dessert ==> Main Course
4	6.15	66.67	1.34	10.83	22.00	Starter & SOFTS & Main Course ==> Dessert
4	6.15	66.67	1.10	10.83	18.00	SOFTS & Menu Wine & Main Course ==> Dessert
4	6.15	64.71	1.34	10.51	22.00	Side Dish & Main Course & COCKTAILS ==> Dessert
3	6.15	63.64	2.13	10.34	35.00	SOFTS & Main Course ==> Dessert
4	6.58	68.00	2.07	10.33	34.00	Steak & SOFTS ==> Starter & Side Dish
4	6.64	68.57	1.46	10.32	24.00	Starter & Side Dish & Dessert ==> Main Course
4	6.89	70.37	1.16	10.22	19.00	Side Dish & Menu Wine & Dessert ==> Steak
3	6.64	67.39	1.89	10.15	31.00	Starter & Dessert ==> Main Course

Figure 43 - Combined Categories By Lift

Inspecting the table of rules sorted by lift it was noted that the rules generated were identical to the rules generated in the R implementation of the same dataset.

Plotting a link graph for the all categories dataset yielded a complex graph with two main clusters around ‘SOFTS’ and ‘Side Dish’ with minor clusters centred around ‘Main Course’ and ‘Starter’ on the right-hand side. This plot was altered to represent the lift value by link thickness, the link colour represents the confidence. Red is higher, blue is lower.

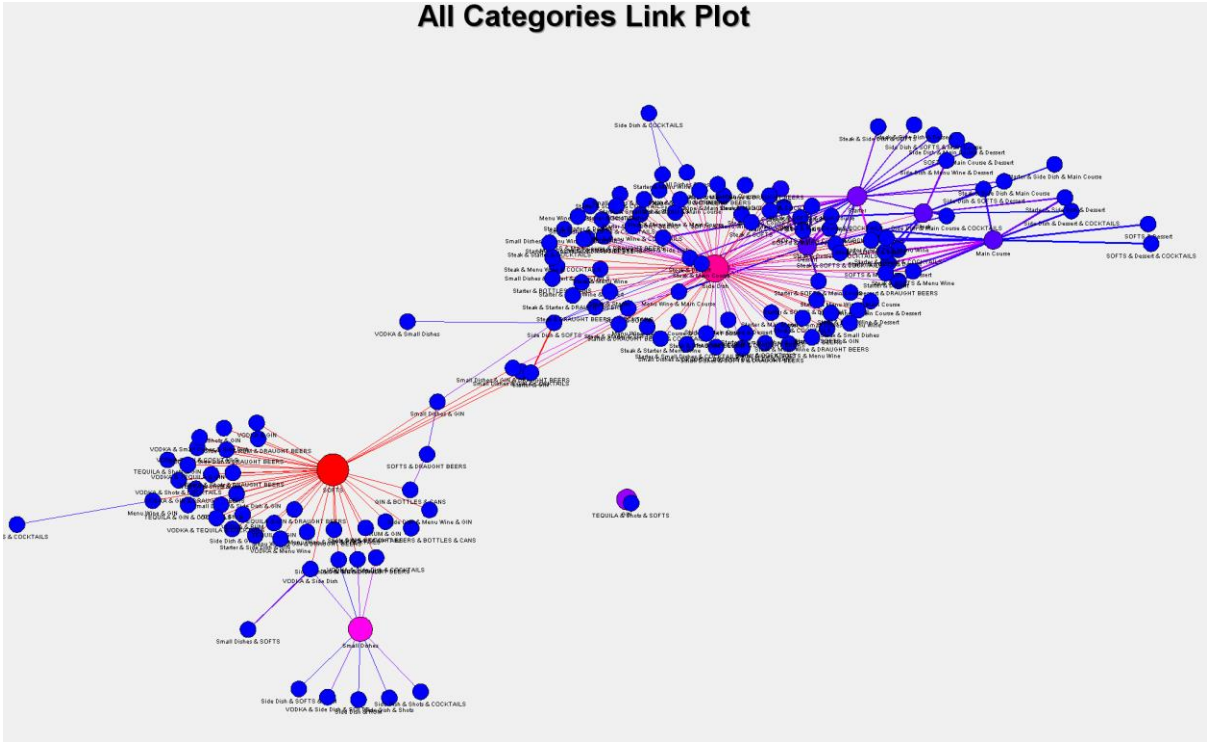


Figure 44- Combined Categories Link Plot

Results Comparison and Conclusion

The initial stages of the analysis across each platform differed. Initial analysis in R was centred around investigating rules generated at an individual product resolution. As this did not generate any rules of interest, this was not repeated in SAS. The second stage of the R analysis modelled the data on a category resolution and did so with all categories in one data frame. The SAS analysis initially considered the food and drinks categories individually and then all together. The only part of the analysis that was conducted across both platforms was the analysis of all categories in a single data frame. This analysis produced identical results, generating the same rules with lift, confidence, and support within a rounding error of each other. This is logical as the 'Apriori' algorithm was implemented identically across platforms: the parameters of minimum support, minimum confidence and rule length were set identically. The only difference between the platforms was the way results are presented. The R 'ruleExplorer' function allows for a user to interact with the rules in order to focus the analysis where as in SAS the 'link graph' is fixed and not adjustable. When many rules are generated, it becomes hard to interpret.

In conclusion both R and SAS implementations were able to mine useful rules from the data. These rules can be further interpreted by subject matter experts to better inform their business strategy and achieve the desired goal of increasing food sales.

REFERENCES

Dunn, P., Allen, L., Cameron, G., Malhotra, M. and Alderwick, H., 2020. *COVID-19 Policy Tracker | The Health Foundation.* [online] The Health Foundation. Available at: <https://www.health.org.uk/news-and-comment/charts-and-infographics/covid-19-policy-tracker> [Accessed 4 January 2021].

Katsigris, C., & Thomas, C. (2012). *The bar & beverage book* (2nd ed., pp. 391-399). Hoboken, N.J.: John Wiley & Sons.

Katsigris, C., & Thomas, C. (2012). *The bar & beverage book* (2nd ed., pp. 399-425). Hoboken, N.J.: John Wiley & Sons.

Wickham, H., & Grolemund, G. (2017). *R for data science*. Beijing . O'Reilly.

Witten, I., & Frank, E. (2017). *Data mining* (4th ed., pp. 120-127). San Francisco, Calif.: Morgan Kaufmann.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining* (3rd ed., pp. 244-271). Amsterdam: Morgan Kaufmann.

Wickham, H (2019). *stringr: Simple, Consistent Wrappers for Common String*

Operations. R package version 1.4.0. https://CRAN.R-project.org/package=stringr

Hahsler, M, Buchta, C, Gruen, B and Hornik, K (2020). *arules*:

Mining Association Rules and Frequent Itemsets. R package version 1.6-6. https://CRAN.R-project.org/package=arules

Hahsler, M (2019). *arulesViz: Visualizing Association Rules and Frequent*

Itemsets. R package version 1.3-3. https://CRAN.R-project.org/package=arulesViz

Wickham, H,François, R. Henry, L and Müller, K (2020*). dplyr: A*

Grammar of Data Manipulation. R package version 1.0.2. https://CRAN.R-project.org/package=dplyr

Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

APPENDICES

Market Basket Analysis R Code.pdf

SAS Data Preprocessing Code.pdf

SAS_data_wrangling.Rmd

Apriori Code Thomas Madeley.Rmd

Double click to open PDF file

Part 3: Can K Means Clustering Meaningfully Group Customer Types Using Behavioural Segmentation?

Introduction

Cluster analysis is a form of unsupervised machine learning that seeks to segment or group data points in subsections or 'clusters'. These data points can have any number of features or measurements and are clustered with data points with a higher similarity than those of other clusters. The main aim of clustering is to identify subsets of data points within a dataset with significantly different properties to those of other groups (Hastie, Tibshirani and Friedman, 2004). This clustering methodology is currently employed by large companies. Banks use clustering to detect fraudulent transactions: Anomaly Detection. The spam filter in your email utilises clustering: Anomaly Detection. Online retailers use clustering to target marketing to your preferences and suggest products you may be interested based on your past purchases: Market Segmentation. This paper seeks to determine if the value that this Market Segmentation methodology affords to large companies can be scaled to a smaller scale and onto a different industry altogether. In this paper a transactional data set from a small hospitality business is investigated using the K Means clustering algorithm to determine whether it can identify transaction types that may offer utility to a small business.

What is K Means Clustering?

Clustering refers to the grouping of unlabelled, sometimes call uncategorised, data. Unlike classification which requires labelled for a model to be trained on and then tested against new unlabelled data. This means that clustering is an 'unsupervised' machine learning methodology. Clustering models use various mathematical mechanisms to create similarity (or dissimilarity) between data points, the data points are then naturally grouped with other points with the highest similarity. This is useful for 2 reasons. Firstly, a large proportion of data is unlabelled and would otherwise be unsuitable for supervised machine learning algorithms. Secondly, that the clusters are naturally formed and not predefined means that the data can be segmented in a way that fits the data best, which may be unexpected or more enlightening. The clustering methodology used in this paper was K-Means cluster. This method first requires a defined number of centroids, K. These centroids are randomly plotted, the data points are then assigned to the nearest centroid using their Euclidean distance to create the first iteration of clusters. The centroid, or 'mean', of these clusters are then calculated. These centroids are then used in the next iteration of clusters. This process continues until the clusters remain unchanged: The clusters have stabilised at the ideal centroids. The method of determining the ideal number of clusters, K, the With in Cluster Sum of Squares (WCSS) is calculated and plotted for a range of values of K. As K increases, the WCSS will reduce sharply and then plateau (Aggarwal, 2015). When plotted on a line plot, the ideal value of K by examining where the reduction in WCSS begins to plateau. This is sometimes called the 'Elbow

Method' as this plateau after the sharp decline forms a visible 'Elbow'. Typically, values between 1 and 10 are plotted in this method. Additionally, the cluster tendency of a data can be defined with the 'Hopkins Statistic'. The Hopkins statistic is a type of hypothesis test where the null hypothesis is: 'The data is uniformly distributed'. Uniformly distributed data, by its nature, will have a very low clustering tendency and a Hopkins Statistic approaching of 0. Data with a high clustering tendency will have a Hopkins statistic approaching 1 (Aggarwal, 2015).

Dataset

The dataset used for this paper comes from a Manchester, UK based bar and restaurant venue called Dive Bar & Grill. Permission for use of the data for research purposes has been granted by the company directors. The dataset contains sales between the 24th September 2020 and 1st November 2020. The dataset is adapted from the raw data from the company's point of sale system which has been previously pre-processed in the Association Rules Mining section of this assignment, full preprocessing steps will be included in the appendix. The previous preprocessing included removing any invalid transactions, any invalid products and removing any sales from outside of the businesses trading times. The pre-processed dataset contains 8 features and 13668 rows. Each row contains an individual product sale with product information and individual transactions can be identified by a common ID number contained within the 'Receipt_ID' column. It is not known if these unique transactions refer to unique customers. Due to the short time interval of the dataset (1 month), it is unlikely that a customer will have visited twice in this time. These unique transactions will therefore be assumed to be unique customers.

Column Name	Format	Description
Receipt_ID	Numeric	Transaction identifying number
Creation_Date	Character	The date of transaction in DD/MM/YYYY format
Name	Character	The name of the product sold
Quantity	Numeric	The quantity of the products sold. Always 1, duplicate products appear on a new row
Tax_Inclusive_Pri	Numeric	The sale price of the product including taxes
Category	Character	The product category to which the product belongs (Draught Beer, Cocktails etc)
Category_Type	Character	The revenue category to which the product belongs. Either 'Food' or 'Drinks'
Course_Number	Numeric	Contains the course for which the product was ordered(Drinks = 0, Starter= 1, Mains = 2, Sides = 3, Dessert = 4)

Figure 45 - Meta Data

Data Preparation and Tools

The main tools used in this paper were R v4.0.3, RStudio v1.3.1093 and SAS Enterprise Miner v14.9. The 'clustertend' v1.4 package was used to test the Hopkins statistic of the dataset. The 'dplyr' v1.02 and 'tidyr' v1.1.2 packages were used to prepare the data. The 'factoextra' v1.07, 'ggplot2' v3.3.2 and 'GGally' v2.0.0 were used to create plots and graphs. SAS Enterprise Miner was used in its base form.

Before beginning to work with the dataset, the required structure of data and features that may assist in producing meaningful customer clusters were considered. Firstly, the individual product sales would need to be grouped into their respective transactions. This is known as pivoting the data from 'long', or one response per row, format to 'wide', multiple responses per row, format. Once the transactions are grouped into individual rows, the total amount spent per transaction would need to be calculated. While the actual products purchased are not important, the number of products purchased per transaction would be an important feature. Whether the customer was a dining or drinking customer is also an important feature. While this binary feature is not ideal for K Means clustering using Euclidean distances, this feature was insisted upon by the client. As the data was only collected over a 5-week period, the date of the transaction would not be an important consideration.

The dataset was first imported into RStudio using the 'read.csv' function. The imported dataset was inspected using the 'head', 'str' and 'nrow' functions to ensure all data had been imported successfully. Using 'dplyr's 'group_by' and 'summarise' functions, a new data frame was created from grouping the the rows by 'Receipt_ID' to capture the individual transactions and summarising by adding all of the 'Tax_Inclusive_Price' values to create a 'Total_Spend' column. As each row in the original data frame represents a product sale, to create a column containing the number of items on each receipt, the 'table' function was used to create a frequency table for each 'Receipt_ID' in original data frame. The frequency column from this table was then written to the grouped data frame. Finally, a logical column containing whether food was ordered was created. This was done first by isolating the 'Receipt_ID' and columns into a new data frame. The 'table' function was used on this data frame to, this resulted in a data frame containing the 'Receipt_ID' and a frequency count column of rows containing 'Food' and a frequency count column of rows containing 'Drinks' grouped by 'Receipt_ID'. Using the 'as.logical' function on the 'Food' column, 0 values were modified to 'FALSE' and values greater than or equal to 1 were modified to 'TRUE'. This 'Food' column was written to the grouped data frame. The resultant data frame contained the receipt id number, the total amount spent on that receipt in Pounds sterling, the number of items in the transaction and whether food items were ordered in the transaction. The data frame contained 1640 transactions and 4 features.

Processed Data Frame

Receipt_ID <int>	Total_Spend <dbl>	Items_Ordered <int>	Ordered_Food <lgl>
93193433	134.00	23	FALSE
98612994	44.70	14	FALSE
104624971	471.15	56	TRUE
106335609	11.50	2	TRUE
106335873	4.30	1	FALSE
106336624	20.35	5	FALSE

Figure 46 – Pre-processed Data

Implementation in R

Firstly, some exploratory analysis was conducted on the pre-processed dataset. To get an understanding of the distributions of the features, histograms for each variable were plotted using 'ggplot2'. The 'Receipt_ID' was not plotted as this is not relevant.

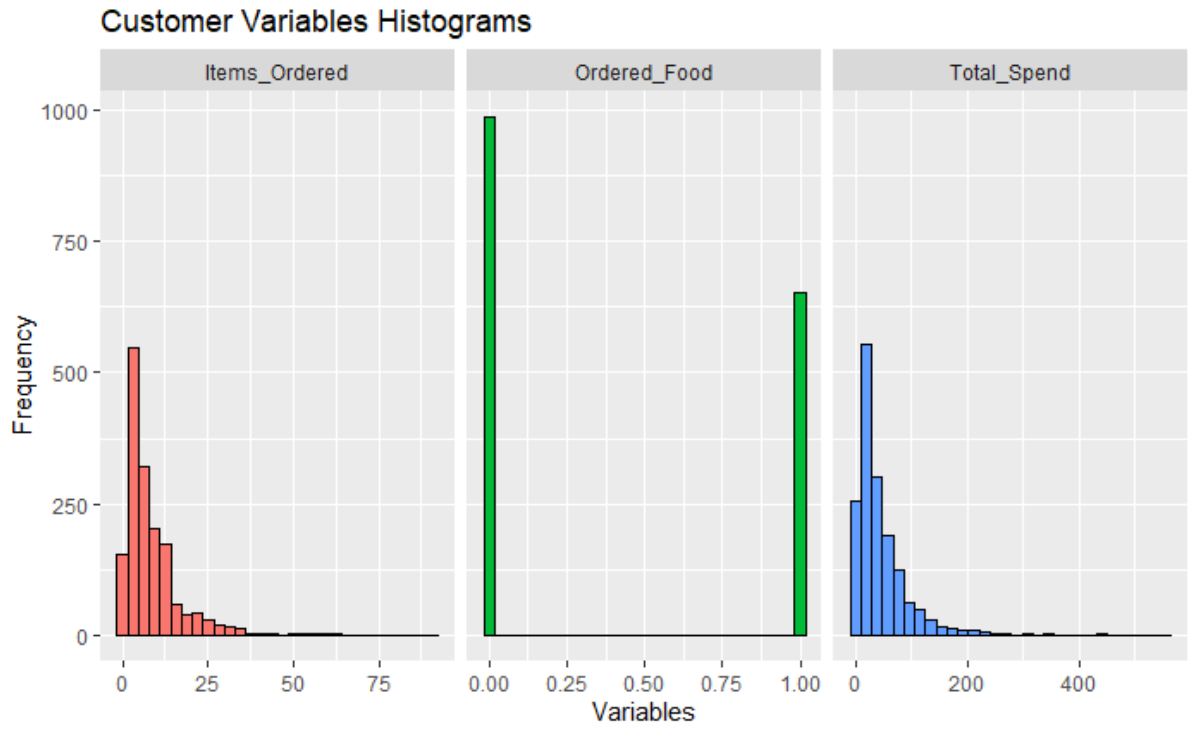


Figure 47 - Variable Distribution Plot

Both the 'Items_Ordered' and 'Total_Spend' are unimodal distributions with a strong right skew. Using the 'summary' function on the data frame showed the mean number of items ordered to be 8.3 and the mean total spend to be 46.09. The number of dining customers was lower than the number of non-dining customers, a further 'group_by' and 'summarise' function was used to determine that there were 987 non-dining and 653 dining customers. Both features had small numbers of outliers. To get a better understanding of the outliers, boxplots were plotted using 'ggplot2'. The boxplots for 'Items_Ordered' and 'Total_Spend' showed narrow interquartile ranges (IQR) for both. They also showed that both features had outliers extending far past the 1.5 IQR limits. This shows that a small number of transactions spend much more than mean and a small number of transactions contain many more products than the mean. Removing of the outliers was considered but ultimately rejected. Due to the nature of the data, these values are not due to instrumental errors and are valid transactions. These 'high spending' may be of predictive value and were therefore retained. It was noted that feature scaling would be required to prevent 'Total_Spend' having a larger influence on the clusters as it has a much larger scale.

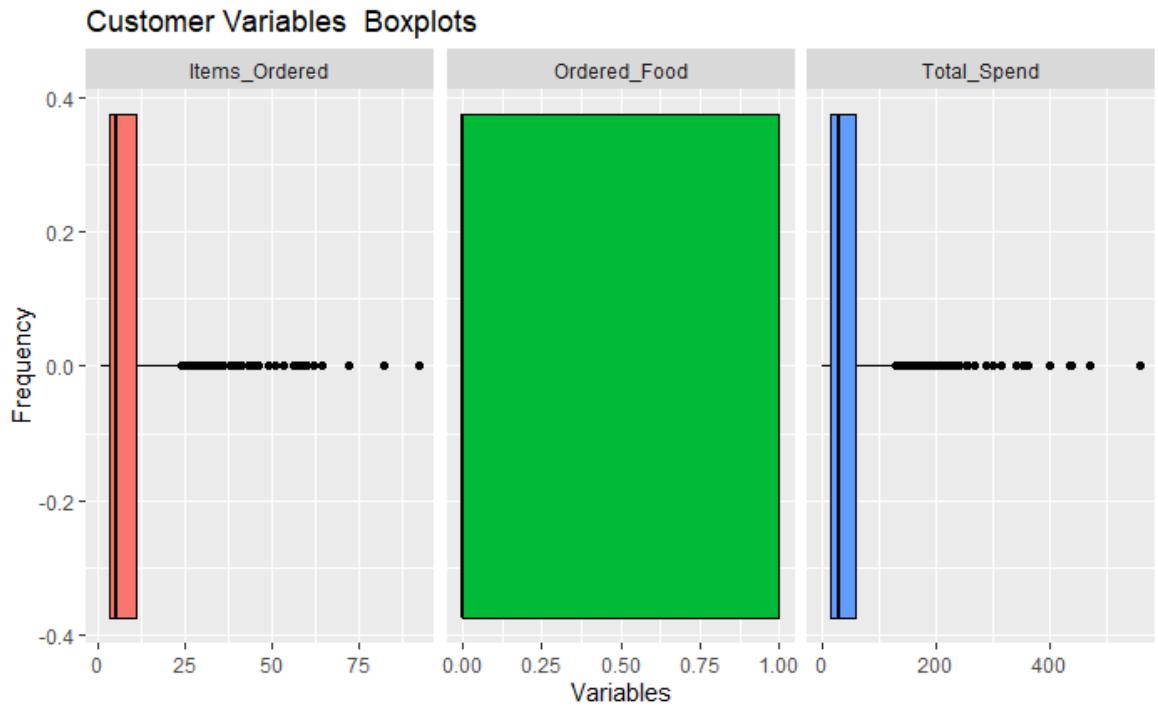


Figure 48 - Variable Boxplots

To understand the correlations between the features, a correlation heatmap matrix was plotted using 'ggplot2'. A correlation matrix displays how correlated a pair of variables are by plotting their correlation coefficients. A correlation coefficient of 1 would indicate a perfect positive correlation, a correlation coefficient of -1 would indicate a perfect negative correlation. A correlation coefficient of 0 would indicate no correlation.

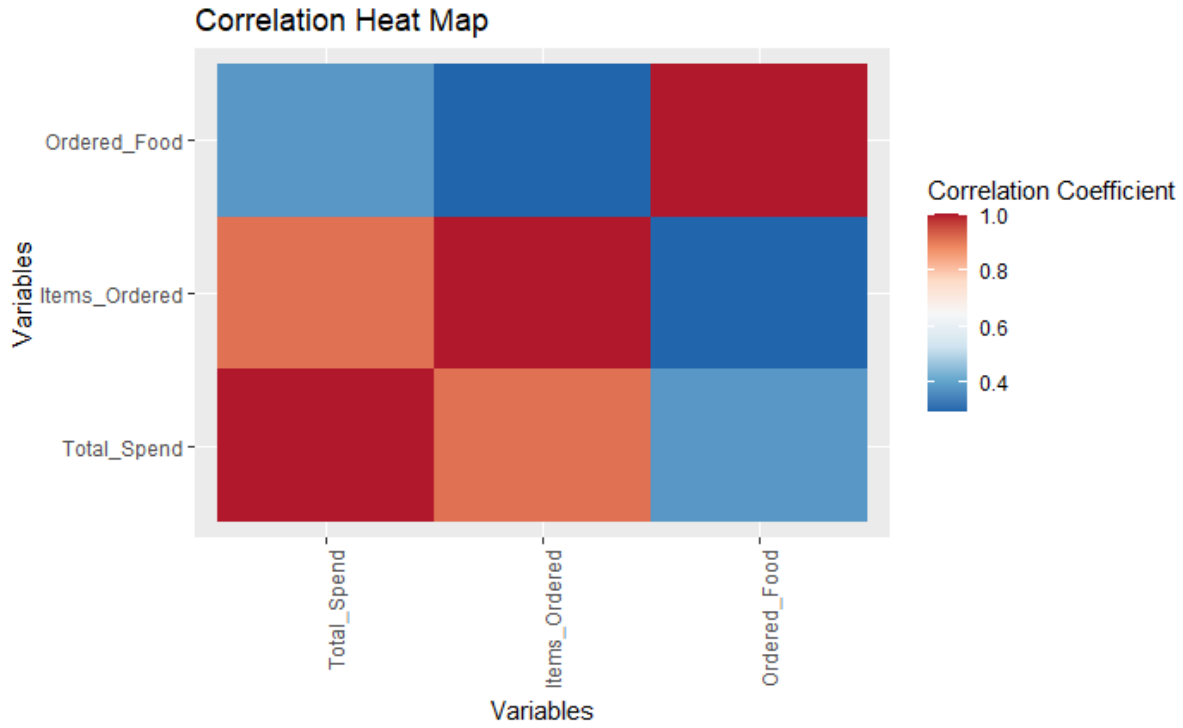


Figure 49 - Correlation Plot

The correlation heatmap matrix shows that 'Items_Ordered' and 'Total_Spend' are highly, positively correlated. This is logical as the amount spent would normally correlate with how many products were purchased. Whether the customer ordered food or not was more strongly, positively correlated with 'Total_Spend' than 'Items_Ordered'. This indicates that dining customers are likely to spend more, which is logical as food items are generally more expensive than drinks items and diners are likely to order drinks as well.

Using the 'factoextra' packages 'get_clust_tendency' function, the Hopkins statistic for the data was calculated to be 0.954. This indicates that there are highly significant clusters within the data.

The data was then standardised. This was done in order to limit the effect of the of variables with larger ranges having a disproportionate effect on the analysis: 'Total_Spend' has a range of over 500 and 'Ordered_Food' has a range of just 1. Z score standardisation was selected over min-max standardisation. This was because of the outliers in both 'Total_Spend' and 'Items_Ordered': Z score standardisation is more able to handle outliers (Aggarwal, 2015). Z score standardisation has the effect of mutating all of the features onto the same scale, normally distributed with a mean of 0 and a standard deviation of 1. Z score standardisation is carried out using the formula:

$$z = \frac{Value - Mean}{Standard\ Deviation}$$

Scaling was implemented using the 'scale' function, the 'center' and 'scale' parameters were set to TRUE to ensure z-score standardisation was carried out. The 'summary' and 'sd' functions were used on each feature to ensure the mean and standard deviation were 0 and 1 respectively. A scatter plot of 'Total_Spend' and 'Items_Ordered' was plotted using 'ggplot2' to visualise the changes. It can be seen that the proportions are the same but both features now share a common scale.

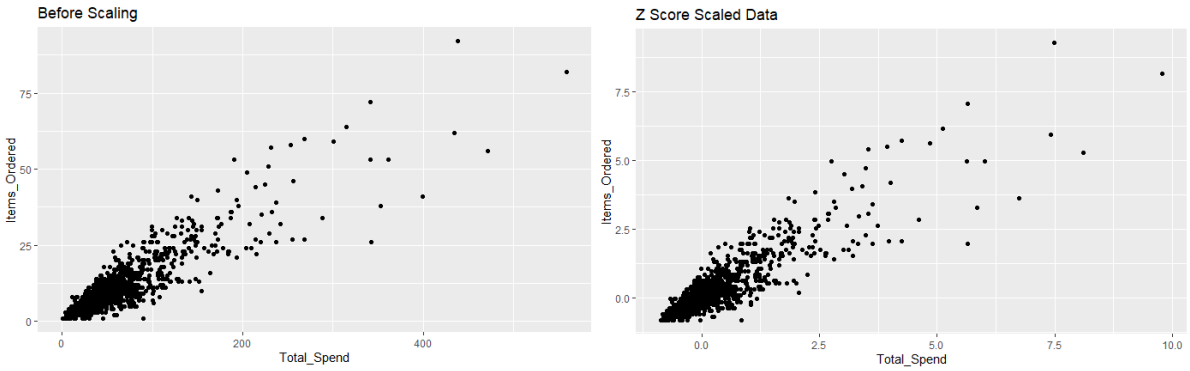


Figure 50 - Scaling Comparison

To test the clustering tendency of the data, the Hopkins statistic was calculated. The Hopkins statistic was calculated to be 0.974 which indicates there are significant clusters in the data.

Before clustering could be implemented, the optimal number of clusters (K) was determined using the 'Elbow Method' (EM). To implement the EM method, so called because of the shape of the resultant graph, the 'Within Cluster Sum of Squares' must be calculated for a range of K values and then plotted. The WCSS was calculated using the 'kmeans' function.

K-Means clustering was then implemented using the 'kmeans' function's 'withinss' function. A function was written to iteratively calculate WCSS for all values between 1 and 10. These were then plotted, and the elbow diagram interpreted. The ideal number of clusters was determined to be 3.

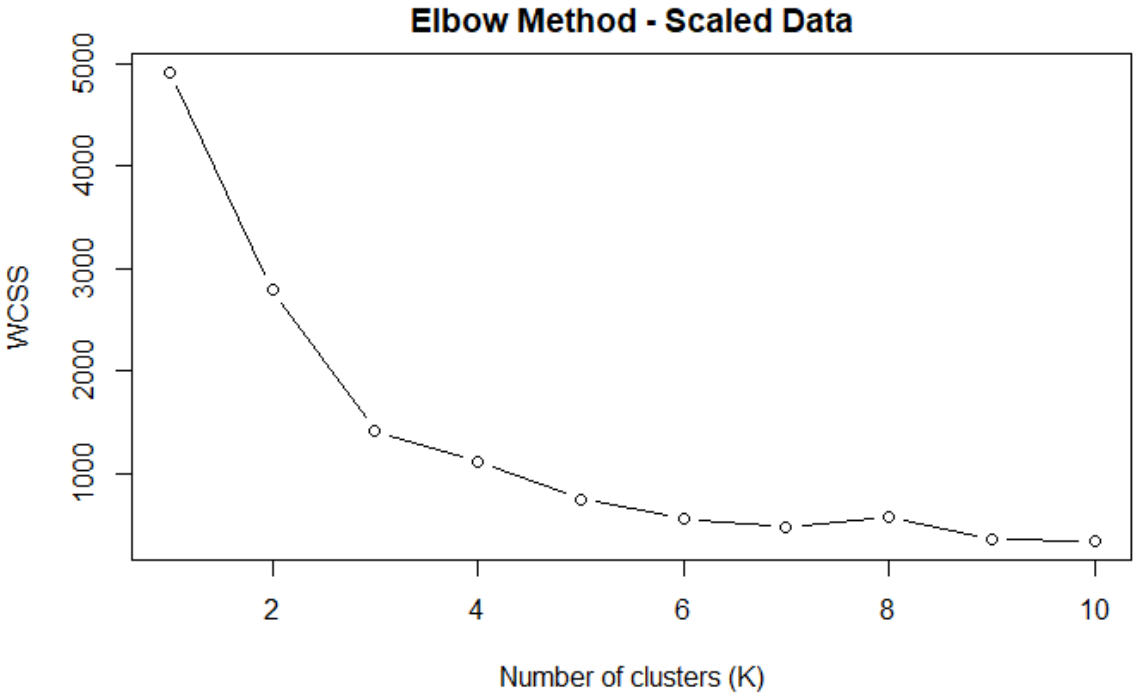


Figure 51 - Elbow Method Diagram

K means clustering was then implemented using the 'kmeans' function and the results were then plotted using the 'factoextra' packages 'fviz_cluster' function.

Results In R

The 'fviz_cluster' function can map these multi dimensional clusters in 2 dimensions by using value decomposition or principal component analysis (PCA). PCA is a linear transformation of the data points to create an eigen value and eigen vector for each variable. The eigen value shows the amount of variance explained by the component, the eigen vector represents the direction of the component : correlation with the other variables. The 2 components that explain the largest variance can be plotted in 2 dimensions in order to get a representation of the clusters (Aggarwal, 2015). Examining the cluster plot from 'fviz_cluster' it can be seen that there are 3 distinct clusters. It was noted that cluster number 3 contains most of the outlier transactions: transactions with a very spend and high number of items purchased. It can also be seen that there is a linear relationship between principal component 1 (PC1) and principal component 2 (PC2). There were also two independent groupings of this linear relationship. The gradient of both groupings is very similar. However, the Y axis intercept is different. There were 558 customers in cluster 1, 956 in cluster 2 and 126 in cluster 3.

Figure 52 - Cluster Visualisation

While these clusters are visually pleasing, without understanding the principal components they are not very informative. To understand the principal components, the 'prcomp' function was used on the scaled data. This resulted in a matrix showing the proportion of variance explained (eigen values) and the rotation (eigen vectors).

```
Importance of components:
                          PC1    PC2     PC3
Standard deviation     1.4551 0.8945 0.28760
Proportion of Variance 0.7057 0.2667 0.02757
Cumulative Proportion  0.7057 0.9724 1.00000
Standard deviations (1, .., p=3):
[1] 1.4550633 0.8944695 0.2876022

Rotation (n x k) = (3 x 3):
                    PC1        PC2         PC3
Total_Spend   -0.6583497  0.2224118 -0.71910268
Items_Ordered -0.6415175  0.3339453  0.69060543
Ordered_Food  -0.3937397 -0.9159768  0.07717171
```

Figure 53 - PCA Results

PC1 and PC2 explain over 97% of the variance cumulatively. This means the cluster plot is a good representation of the clusters despite the dimensional reduction: adding a third dimension would only explain an additional 3%. The principal components were then plotted in a biplot using the 'factorextra' packages 'fviz_pca_biplot' function. This plot displays how each variable relates to the principal components by the eigen vectors for each variable.

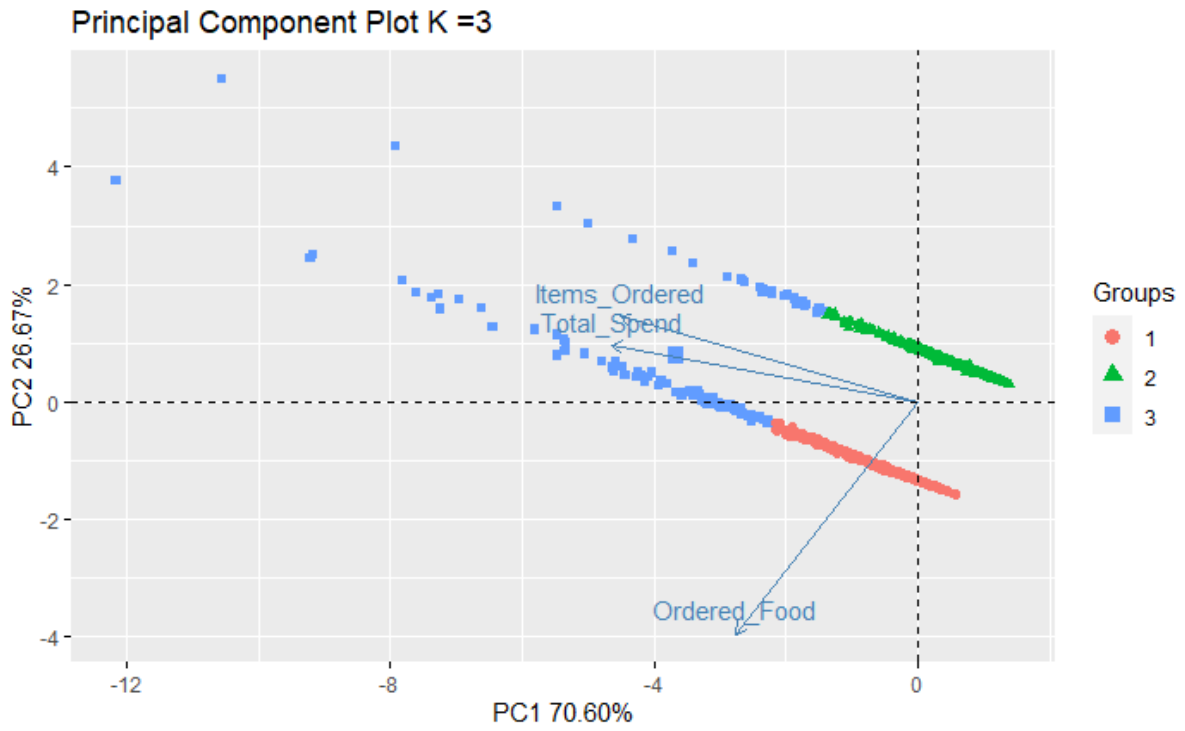


Figure 54 - Principal Component Plot

With the eigen vectors overlayed it is clear that 'Items_Ordered' and 'Total_Spend' have a strong influence on the PC1 axis and a small influence on the PC2 axis as they have a large, negative, horizontal component and a small, positivel vertical one. 'Ordered_Food' has strong influence on the PC2 axis as it has a large, negative, vertical component.

Now that the principal component axes are understood, some conclusions can be drawn from the clusters. The two distinct linear groups represent customers dining and non-dining customers, with non-dining customers being the lower of the two linear groups. Cluster 1 therefore contains dining customers with a low number of items ordered and a low total spend. Cluster 2 contains non-dining customers with a low number of items ordered and a low total spend. Cluster 3 contains the outlier customers with both a high number of items ordered, and high total spend and includes both dining and non-dining customers. To finalise the analysis a pairwise plot of the variables was conducted, the data points were coloured in to show their assigned cluster.

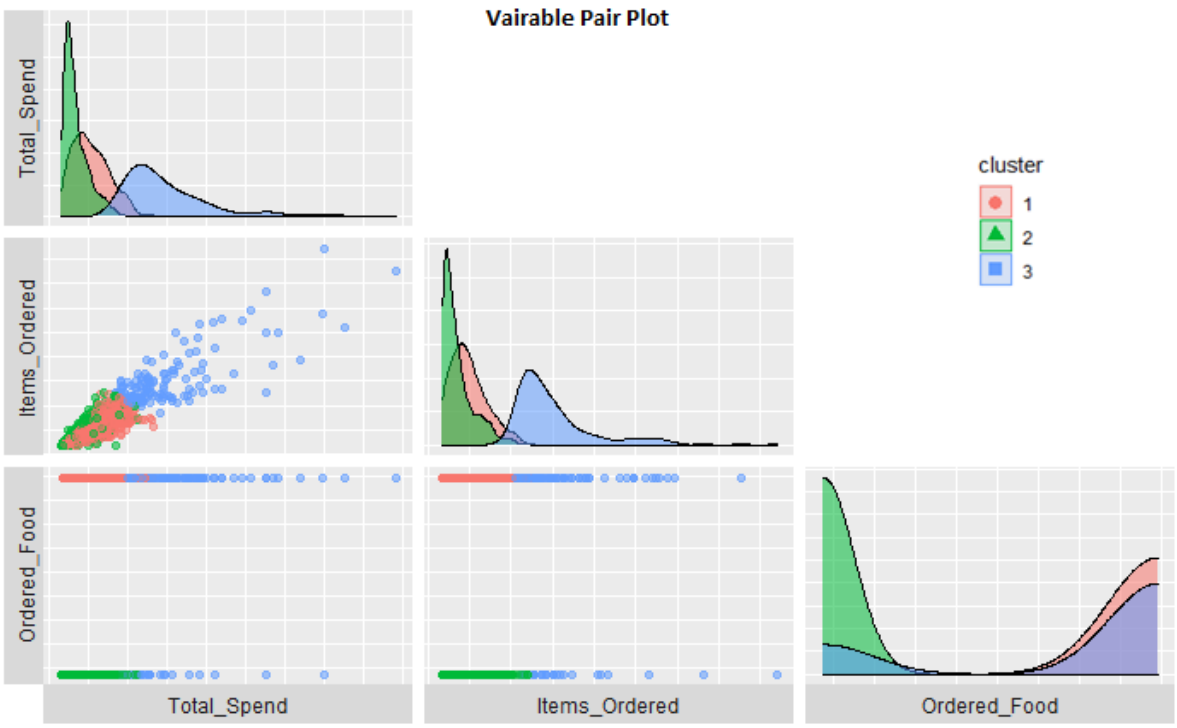


Figure 55 - Variable Pair Plot

In summary, three distinct customer groups were identified from the K means cluster and they can be described as: Cluster 1 – Dining Customers, Cluster 2 – Drinking Customers, Cluster 3 – High spending customers.

SAS Implementation

The SAS implementation was relatively simple in comparison to R. The 4-feature pre-processed dataset was first written to CSV in R studio. A diagram was then created in SAS and then the data was imported into SAS using a 'File Import' (FI) node. The FI node was run and the imported data was explored. Using the 'results' tab of the FI node, the distributions of both 'Total_Spend' and 'Items_Ordered' were plotted.

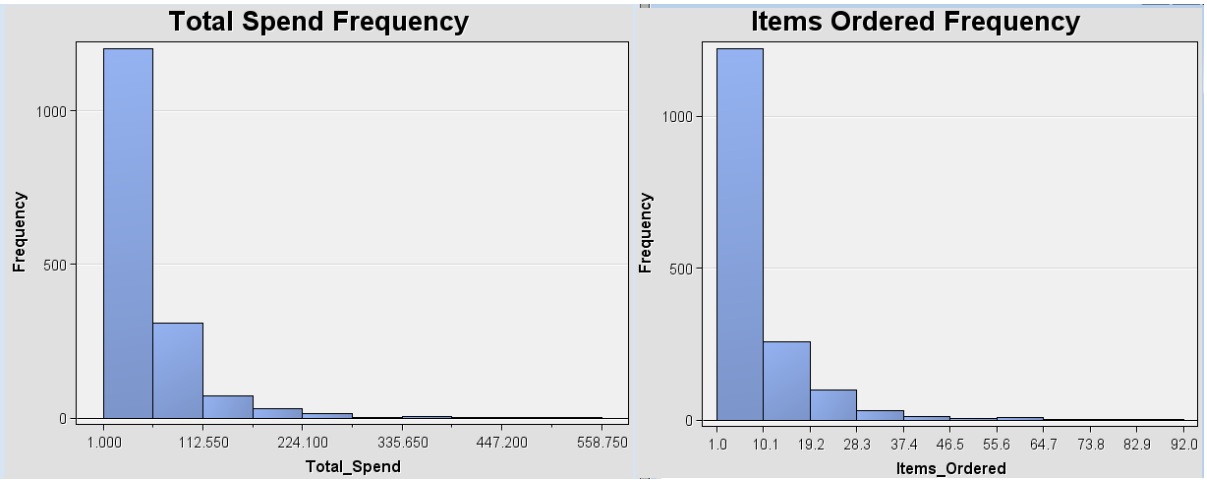


Figure 56 - SAS Variable Frequency Plot

The distribution matches the distribution found in R, however the binwidth in SAS is wider and it therefore visually appears slightly different. The 'Receipt_ID' Column was dropped using the FI node's 'Variables' tab.

A 'Cluster' node was then added to the diagram. The 'Internal Standardisation' method within the cluster node was set to 'Standardisation'. The method used is SAS differs to the method z-score method used in R. The SAS standardisation does not subtract the mean from the value and only divides by the standard deviation which may produce different results:

$$SAS\ Standaradisation = \frac{Value}{Standard\ Deviation}$$

(53805 - Cluster node standardization when the Number of Clusters Specification Method is Automatic, n.d.)

This was due to the outliers highlighted in the the R implementation. The number of clusters was set to 'Automatic'. This was done to investigate the number of clusters generated by SAS. The clustering method was set to 'Centroid'. This defines the distance between two clusters as the squared Euclidean distance between the centroids. The 'Cluster' node was then run. Finally, a 'Segment Profile' node was added to allow for more in depth analysis of the clusters.

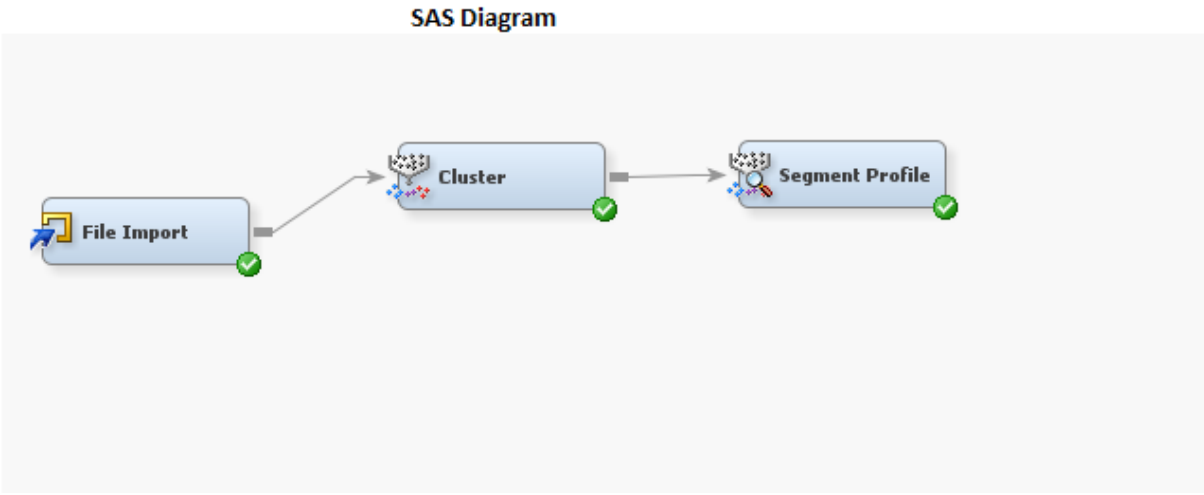


Figure 57 - SAS Process Flow Diagram

Results in SAS

The results of the 'Cluster' node were inspected first. The node automatically generated 3 clusters containing 575, 961 and 104 observations, respectively.

Viewing the 'Segment Plot' whether a customer dined or not was highly significant in determining which cluster an observation would belong to: Cluster 1 and 2 are divided by this feature with Cluster 1 containing only customers that dined and Cluster 2 containing only customers that did not dine. Cluster 3 contained both dining and non-dining customers. Examining the 'Segment Plot' for 'Total_Spend' the majority of Clusters 1 and 2's customers spent less that £70: 73% of customers in Cluster 1 and 94% of Cluster 2 spent less that £70. Both Cluster 1 and 2 contained a small number of customers spending between £70 and £140. Cluster 3 consisted of approximately 27% of customers spending between £70 and £140, 43% of Cluster 3 customers spent £140 and £210 and continually smaller minorities spent more than this up to the top 1% of customers spending £490 - £560. The 'Items_Ordered' Segment plot shows very high correlation to 'Total_Spend': Both contain a large proportion within a low range, with a minority contingent in a higher range.

Figure 58 - SAS Segment Plot

Examining the mean statistics, it can be seen that the mean 'Total_Spend' for Cluster 2 (non-dining customers), was the lowest and the mean for Cluster 1 (dining customers) was nearly double despite the mean 'Items_Ordered being only slightly higher (5.37 and 8.63 respectively). Cluster 3 appears to contain all the higher spending customers and is made up of a majority of dining customers.

Segment Id ▲	Frequency of Cluster	Total_Spend	Items_Ordered	Ordered_Food=TRUE	Ordered_Food=FALSE
1	575	52.52774	8.627826	1	-7.5E-15
2	961	26.19012	5.37461	-4.6E-16	1
3	104	194.4327	34.05769	0.75	0.25

Figure 59 - SAS Mean Statistics

The results tab of the 'Segment Profile' node was used to plot the 'Profile Segment Plot'. This plot showed the distribution of variables within in a cluster as a proportion of the total distribution. The distribution within the cluster is plotted in a solid colour and the total distribution is skeletonised in an overlay so that a comparison can be made. This plot shows that the dining customers in Cluster 1 are ordering less products and have a more consistent total spend. A large number non-dining customers in Cluster 2 are ordering a very small number of products and spending very little also.

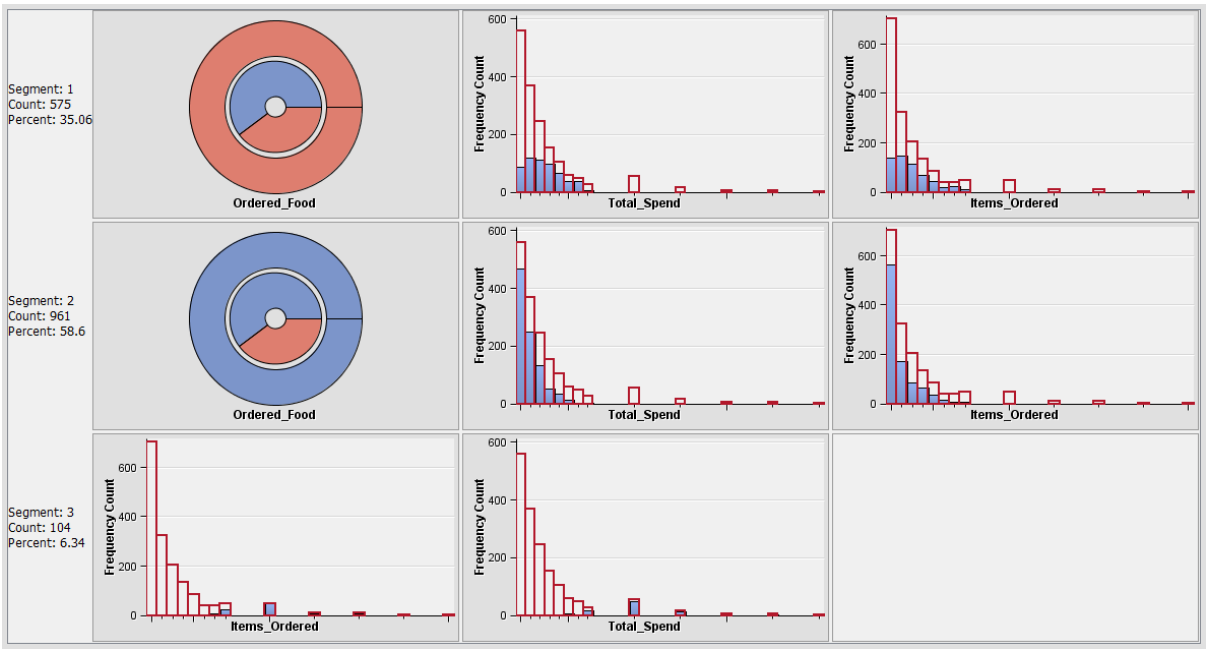


Figure 60 - SAS Segment Profile

Results Comparison and Conclusion

SAS and R achieved very similar results in this analysis. Both platforms' methods generated 3 clusters and both platforms' main 2 clusters divided the customers into dining and non-dining groups. The 'Variable Pair Plot' from R and the 'Profile Segment Plot' show these similarities. To further highlight the similarities, the 'Principal Component' node was used to replicate the Principal Component plot that was done in R, the plot is inverted relative to the R plot.

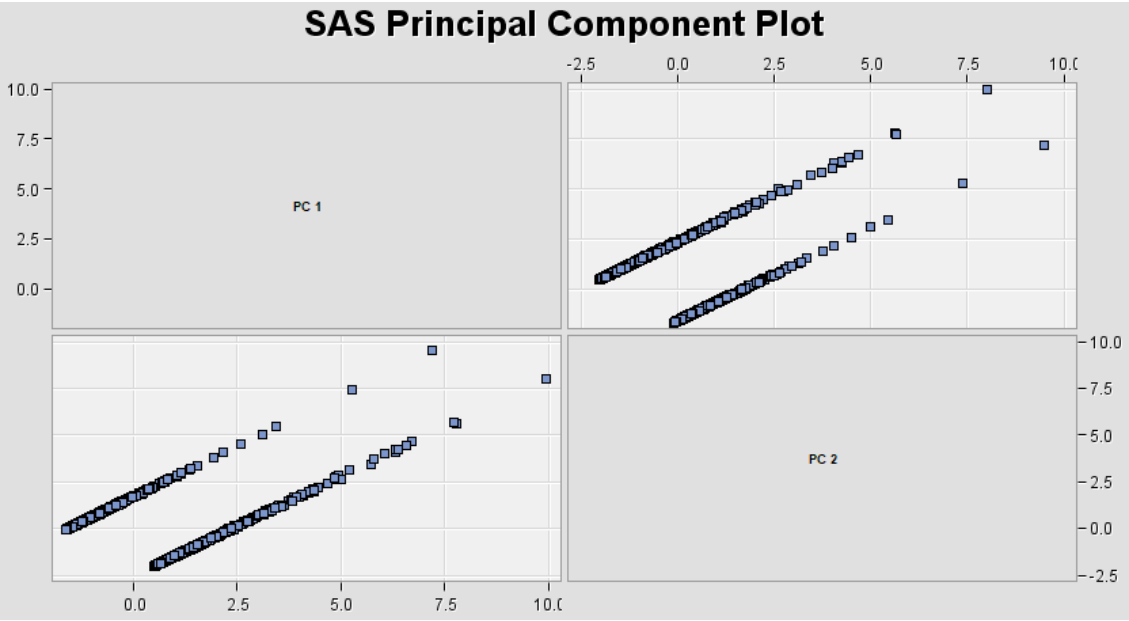


Figure 61 - SAS Principal Component Plot

The main difference in the results was the number of customers clustered into Cluster 3. R Clustered 126 customers into Cluster 3 and SAS clustered only 104. As Cluster 3 consisted of high spending dining and non-dining customers, this means that the ceiling value to be clustered into Cluster 3 in SAS 'Total_Spend' and 'Items_Ordered' in SAS was higher. This is due to the difference in standardisation techniques across the two platforms. In R, z-score standardisation was carried out. This led to the standardised data having a mean of 0 and a standard deviation of 1, effectively transforming the data distribution to a normal distribution. In SAS, the standardisation technique simply divided by the standard deviation. This means that the data would not be 'centered' around 0 and would retain its original mean: Mean / Standard Deviation. That the new means for all features would be greater than 0, and therefore greater than the mean in R, the data would be shifted along the x axis relative to the R data.

In conclusion, both platforms were able to generate meaningful clusters that can better inform the venue about their clientele and allow them to directly market to specific customer groups in future based on their past preferences.

REFERENCES

Hastie, T., Tibshirani, R. and Friedman, J., 2004. *The Elements Of Statistical Learning.* 2nd ed. New York: Springer, pp.485 - 490.

Hastie, T., Tibshirani, R. and Friedman, J., 2004. *The Elements Of Statistical Learning.* 2nd ed. New York: Springer, pp.505 - 515.

Support.sas.com. n.d. 53805 - *Cluster Node Standardization When The Number Of Clusters Specification Method Is Automatic.* [online] Available at: <https://support.sas.com/kb/53/805.html> [Accessed 8 January 2021].

Aggarwal, C., 2015. *Data Mining*. New York: Springer, pp.157 - 158.

Aggarwal, C., 2015. *Data Mining*. New York: Springer, pp.37.

Aggarwal, C., 2015. *Data Mining*. New York: Springer, pp.43

APPENDIX

Clustering Code Thomas Madeley.pdf

Clustering Thomas Madeley.Rmd

Part 4: Sentiment Analysis: Using Natural Language Processing to Compare Destinations

Abstract

Sentiment analysis is the study of human opinions and emotions from natural language text using computational methods. Scientific interest in sentiment rose in unison with the rise of digital user generated content. This user content, such as reviews, online forums, social media posts can be of value to businesses (Liu, 2015). For example, a business may analyse product reviews in order to guide develop their next product (Liu, 2015). Due to the enormous volume of content created, it is unfeasible to analyse manually. Computational methods can analyse thousands of text documents and output linked topics, classify the sentiment contained within or even group customers together based on similarities in their opinion. This paper explores a dataset of Thai hotel and restaurant reviews and seeks to discover whether sentiment analysis can yield useful information from past reviews, that may be used to influence the location choice of potential future visitors.

Introduction

Humanity is accumulating data at an ever-increasing rate. At the time of writing 90% of the data accumulated by humanity was accumulated in the last two years ("How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read", 2020). This rapid growth of data is largely due to the rise of the internet and with it, social media, online shopping and web searches. Classical machine learning models such as Classification, Regression or Clustering are essentially mathematical functions and require numerical input. Unfortunately, a large proportion of this data is text. For example, Twitter users tweet 473,400 tweets per minute and Google processes 40,000 searches every second ("How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read", 2020). In order to make use of this data, Statistical Natural Language Processing (NLP) must be utilised. One subfield of NLP is Sentiment Analysis (SA). Humans are excellent at analysing sentiment and we do it in our daily lives. One example of this is reading the reviews of a product before we make a purchase. We analyse the sentiment of other customers' reviews in order to inform our purchase. In this paper, SA was implemented on an unlabelled dataset of Thai hotel and restaurant reviews in order to extract insights and statistics that may be used to better inform customers. The most popular destination and its geographically closest neighbour will be selected. Ten hotels from each location were selected. The reviews from both the locations will be analysed using NLP and SA to determine: Can NLP techniques and SA of past reviews be used to mine information that can influence a customer's choice of holiday destination?

Data, Tools Used and Data Pre-processing

The dataset used in this paper was provided by the University of Salford, School of Computing, Science and Engineering. The dataset contains 53644 hotel and restaurant reviews from 537 different venues and 25 different locations in Thailand. The dataset contains 5 features, all of character format. The reviews are mostly in English with some foreign languages being used also.

Column Name	Format	Description
ID	Character	Unique identifying code for each review
Hotel/Restaurant Name	Character	Name of venue
Location	Character	The town in which the venue is located
Review Date	Character	Contains the date the review was posted. If posted recently it states "X days ago"
Review	Character	Contains the text of the review. Some non english reviews and special characters.

Figure 62 - Meta Data

The main tools used in this paper are R v 4.03, RStudo v1.3.1093 and SAS Enterprise Miner V14.3. Within RStudio, 'ggplot2' v3.3.2 and 'ggthemes' v4.2.0 were used to create graphs and plots. 'dplyr' v 1.0.2 was used for data transformation. 'tm' v 0.7-8 Was used to prepare the text data. Other notable packages are 'wordcloud' v2.6, 'tidyverse'v1.3.0, 'tibble' v 3.04 and 'stringr' v1.4.0.

Data Pre-processing and Hotel Selection

The dataset provided was initially in Microsoft Excel format (.xls). This was first opened with Microsoft Excel and then saved into comma separated value format(.csv). All further data pre-processing steps were completed in RStudio. The new .csv file was imported into RStudio using the 'read.csv' function. Using the 'dim', 'head' and 'str' functions the data was inspected. The number of rows and columns was determined. It was noted that the dataset is unlabelled and that each row contained a review with 4 identifying features: An identifying ID code, the name of hotel or restaurant and the location of the hotel or restaurant. The 'unique' function was used on the 'Hotel.Restaurant.name' column in order to determine the number of unique venues in the dataset (537).

Initially it was decided that the criteria for hotel and location selection was to be the hotels with the greatest number of reviews within the locations with the largest number of hotels. Using the 'group_by' and 'summarise' function, it was determined that 379 of the 537 venues had 100 reviews exactly with a further 2 having over 100 reviews. Using the 'filter' function on the grouped data frame above, the data for all of the hotels with at least 100 reviews was written to a new data frame. The 'Hotel.Restaurant.name' column from this data frame was then written to a new variable to create a list of all hotel names with over 100 reviews. This list of names was then used with the 'filter' function on the original dataset to create a new data frame containing only entries for venues with at least 100 reviews. Using a further 'group_by' and 'summarise' function, the entries were grouped by location and summarised by the venue name. This created a table showing the location and the number of venues with over 100 reviews. 'Patong' was identified as the location with the

largest number of venues with at least 100 reviews (123 venues) which was interpreted to mean it was the most popular destination. The next location was selected based on geographical proximity to 'Patong'. Using Google Maps it was determined that 'Karon' are 'Kamala' are both equidistant from 'Patong'. 'Karon' was ultimately selected as it has more venues with over 100 reviews (39 to 21 respectively). The remaining data frame was then filtered using the 'filter' function to create 2 separate data frames for both the 'Patong' reviews (patong_reviews) and 'Karon' reviews (karon_reviews). It was noticed that the location column contained white space before the location name, this was removed using the 'trimws' function. To select the required 10 venues from each location the, the mean length of each venue's reviews was calculated, then arranged in descending order. The 10 venues with the longest mean review length's names were then written to a new list.

Calculating mean review length

```
top_patong_length<-patong_reviews %>% group_by(Hotel.Restaurant.name, Location) %>%
summarise(mean_length = mean(nchar(Review))) %>% arrange(desc(mean_length))
top_karon_length <- karon_reviews %>% group_by(Hotel.Restaurant.name, Location) %>%
summarise(mean_length = mean(nchar(Review))) %>% arrange(desc(mean_length))

top_patong_length <- head(top_patong_length,10)
top_karon_length<- head(top_karon_length, 10)

top_patong_names <- top_patong_length$Hotel.Restaurant.name
top_karon_names <- top_karon_length$Hotel.Restaurant.name
top_karon_length
top_patong_length

```

Hotel.Restaurant.name <chr>	Location <chr>	mean_length <dbl>
Tunk-Ka Cafe	Patong	241.37
Sizzle Rooftop Restaurant	Patong	233.54
K-Hotel Restaurant and Beer Garden	Patong	232.77
Climax on Bangla	Patong	231.67
Poo Nurntong Restaurant	Patong	231.15
Kokosnuss	Patong	231.00
Baan Rim Pa Patong	Patong	230.28
La Gritta	Patong	229.55
La Dolce Vita Restaurant	Patong	229.02
Ao Chalong Yacht Club Restaurant	Patong	228.90

Figure 63 - Calculating Mean Review Length Code

The lists of names were then used to filter both the Patong reviews and Karon reviews, creating new data frames containing only 10 venues from both locations containing the largest mean review length, containing 1000 reviews each. Both data frames were then checked to confirm that the venues were 'active': received a review within 4 weeks of the data being captured as analysing a venue that is no longer active or closed would not be useful. The 'Review.Date' column was in character format. It was noted that recent reviews showed a date of 'Reviewed X days ago' if they were newer than 4 weeks ago. To circumvent the character format of the data, the 'stringr'

package's 'str_detect' function to search for a partial string match within the 'Review.Date' column. Using the 'summarise' function, entries containing 'ago' would return an 'active' status.

Checking If Venues Are Active

```
library(stringr)
active_status_patong <- patong_reviews_10 %>% group_by(Hotel.Restaurant.name) %>%
summarise(Active = str_detect(Review.Date, "ago")) %>% summarise ("Active Status" =
any(Active))

active_status_patong

active_status_karon <- karon_reviews_10 %>% group_by(Hotel.Restaurant.name) %>%
summarise(Active = str_detect(Review.Date, "ago")) %>% summarise ("Active Status" =
any(Active))

active_status_karon
```

Hotel.Restaurant.name <chr>	Active Status <lgl>
2gether Restaurant	FALSE
El Gaucho Steakhouse	TRUE
Ging Restaurant	TRUE
Kalika 76 Restaurant	TRUE
O-Oh Farm Karon	TRUE
PaPa Restaurant	TRUE
Sabaijai Cafe	TRUE
Schlusslicht	TRUE
Tandoori Flames	TRUE
The Palm Cuisine	TRUE

Figure 64 - Confirming Active Venue Code

As one of the venues from the Karon data frame contained 'FALSE' it was deemed inactive. This venue was removed from the final selection of 10 and the venue with the 11th longest mean review length was added instead. Once the process was repeated and rechecked the final selections of 10 venues for each location were written to CSV format for future use in R and SAS. Finally, using a lapply function on the data frames for both locations, each individual venue was written to its own CSV file so that the venues may be analysed individually.

Function For Creation Of Unique CSV Files

```
library(tidyverse)



# Split by Hotel.Restaurant.name
patong_split <- split(patong_reviews_10, patong_reviews_10$Hotel.Restaurant.name)

# Saving them as a csv with a comma seperator
lapply(names(patong_split), function(x){
    write_csv(patong_split[[x]], path = paste(x, ".csv", sep = ","))
    })



# Split by Hotel.Restaurant.name
karon_split <- split(karon_reviews_11, karon_reviews_11$Hotel.Restaurant.name)

# Saving them as a csv with a comma seperator
lapply(names(karon_split), function(x){
    write_csv(karon_split[[x]], path = paste(x, ".csv", sep = ","))
    })

```

Figure 65 - Function To Write CSV Files

Implementation in R

In order to conduct sentiment analysis on the data frames created in the pre-processing steps, they must be parsed into corpora. To do this several further steps were taken. All the pre-processed data frames were imported with the 'read.csv' function. Using the 'tolower' function on the 'Review' column, all the reviews were converted to lower case. Using the 'gsub' function and regular expressions, text unnecessary to the analysis was removed: URLs and hyperlinks, punctuation, numeric digits and white space were all removed. The reviews where then converted into corpora using the 'Corpus' function. Using the 'inspect' function to examine the created corpora, it was noted that there were several line breaks ("\n") within the text. There were also several non-

alphanumeric characters within words, such as “ãâ,â„”. The ‘gsub’ function was further used to remove these. The corpora were then rechecked. Using the ‘tm’ packages ‘tm_map’ function to transform the corpora, English language stop words were removed such as ‘is, the, and not’. Stop words are generally not useful for sentiment analysis as their frequency can skew results. A potential drawback of this is loss of context for some words, for example ‘not’ can alter meaning of a word placed after it. To reduce the words within the corpora to their root form, the corpora were stemmed using the ‘tm_map’ function in conjunction with the ‘stemDocument’. Stemming reduces the complexity and dimensionality of the text by removing prefixes and suffixes from words. This reduces the word to a simpler form called a ‘stem’: Word = “Eating, ”, Stem = “Eat”.

In order to conduct sentiment analysis, a positive and negative lexicon are required against which to compare the words in the corpora. Lexicons from the ASDM workshop were imported. A function to conduct sentiment analysis, (named Sentiment) was adapted from a function obtained from a University Workshop (2020, University of Salford). The function first creates a wordcloud using the ‘wordcloud’ package. Then uses the intersect function to return a numerical value of intersects between the corpus and the lexicons. This returns a count of both how many positive words and how many negative words are in each corpus. The percentage of positive to negative words in each corpus is then returned. This method was considered adequate for an overview of the sentiment in a location. However, it may transpire that a small number of reviews contain many positive words or negative reviews containing many negative words, which could skew the results.

To understand the number of positive and negative reviews, the Sentiment function was adapted to classify the reviews rather than count the overall words. The function was named ‘sentiment_class’. This was done by inserting an ‘if/else’ statement that would compare the positive and negative counts, adding 1 to the positive review count if there were more positive words than negative words, and adding 1 to the negative review count if there were more negative words than positive words. This ‘sentiment_class’ prevents reviews containing many either positive or negative reviews to skew the overall sentiment towards a venue or location and allows the quantity of positive or negative reviews to be analysed instead.

Review Classifier Function

```
sentiment_class <- function(stem_corpus)
{

#Calculating the count of total positive and negative words in each review

#Create variables and vectors
total_pos_review <- 0
total_neg_review <- 0
#pos_count_vector <- c()
#neg_count_vector <- c()
#Calculate the size of the corpus
size <- length(stem_corpus)
for(i in 1:size)
{
 #All the words in current review
 corpus_words<- list(strsplit(stem_corpus[[i]]$content, split = " "))
 #positive words in current review

 pos_count <-length(intersect(unlist(corpus_words), unlist(positive_lexicon)))
 #negative words in current review
 neg_count <- length(intersect(unlist(corpus_words), unlist(negative_lexicon)))

 if (pos_count > neg_count)

 total_pos_review <- total_pos_review + 1

 else
 total_neg_review <- total_neg_review +1

}
#Calculating overall percentage of positive and negative  reviews
total_pos_review ## overall positive count
total_neg_review ## overall negative count
total_count <- total_pos_review + total_neg_review
overall_positive_percentage <- (total_pos_review*100)/total_count

overall_positive_percentage ## overall positive percentage
#Create a dataframe with all the positive and negative reviews
df<-data.frame(Review_Type=c("Postive","Negative"),
 Count=c(total_pos_review ,total_neg_review ))


overall_positive_percentage<-paste("Percentage of Positive Reviews:",
round(overall_positive_percentage,2),"%")
print(overall_positive_percentage)

return(df)
}
```

Figure 66 - Sentiment Classifier Function

Results in R

Initially the results for the analysis of positive words in each location was analysed using the 'Sentiment' function. The results for both locations were very similar, Patong received 85.68% positive words in its reviews, Karon received 85.63%. It was noted that the word clouds for both locations were also very similar.

Figure 67 - Karon Reviews Wordcloud

Figure 68 - Patong Reviews Wordcloud

To gain further insight into the commonality of the most common words used across the locations, the 'TermDocumentMatrix' function was used to calculate the frequency of each word use, the 20 highest frequency words from each location were plotted on a side by side bar plot.

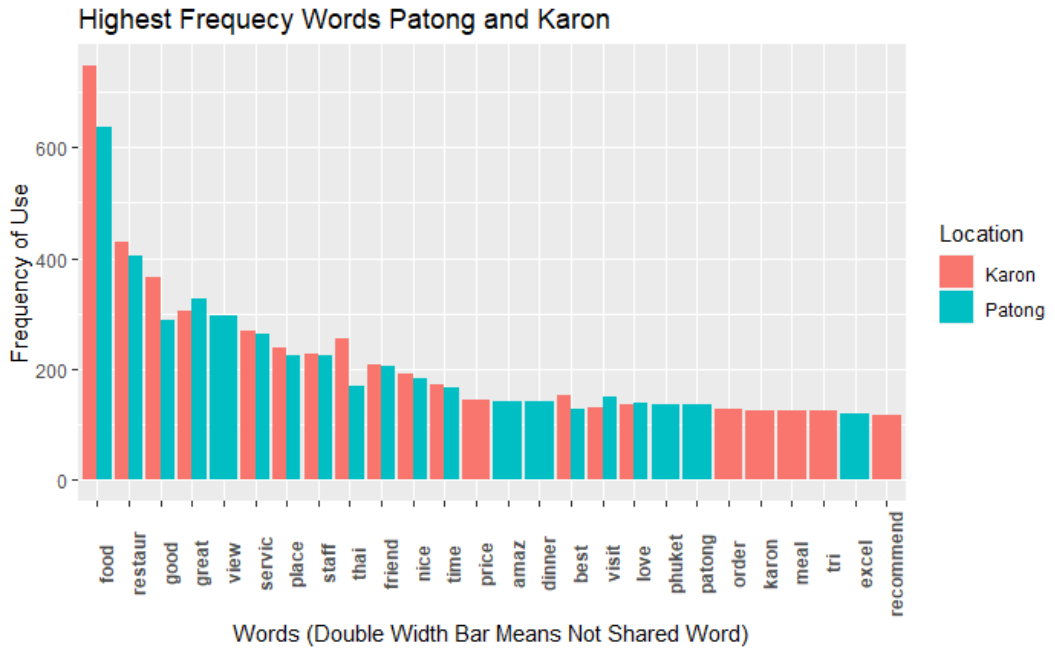


Figure 69 - Word Frequency Across Destinations

It was noted that not only was there high commonality between the locations, but there was also a very similar frequency of use of most words. Of the top 20 words from both locations, 65% were shared across both locations. While some of the words are unique in the top 20 by frequency for

each location, we cannot discount that they are simply further down the list rather than not mentioned at all. Extrapolating the differences in most frequently used words across both locations it can be stated that: The view in Patong is more noteworthy, for better or worse. Customers are more likely to mention the price in Karon, for better or worse. Customers are more likely to recommend a venue in Karon. Examining the frequency of the common word: Customers are more likely to mention the food in Karon. Customers are more likely to rate a venue as 'Great' in Patong and 'Good' in Karon. Karon may offer a more authentic 'Thai' experience.

The results from the 'Sentiment' function when applied to the individual venues was then analysed. Using the 'ggplot2' boxplot the percentages of positive words per venue were plotted.

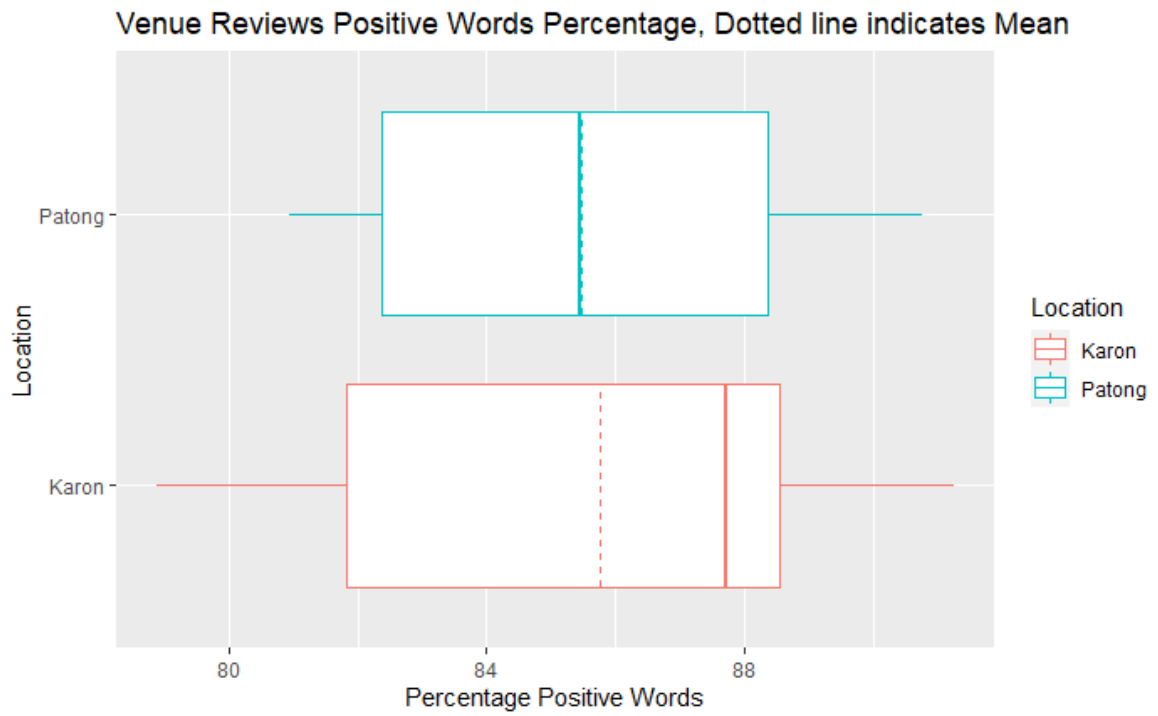


Figure 70 - Boxplot, Postive Words %

It was noted that the Patong ratios of positive to negative words were normally distributed with an approximately equal mode and mean. The Karon ratios are however significantly left skewed. This would indicate that over a prolonged stay in Karon, a customer may experience a better quality of venue *on average.* However, it should be noted that as the interquartile range (IQR) and range are larger in Karon, a customer is also more likely to have a single worse experience. A prolonged stay in Patong may result in, on average, a more consistent quality of venue.

This analysis was also carried out on the positive and negative reviews, as classified by the 'sentiment_class' function outlined above: Number positive words > number negative words.

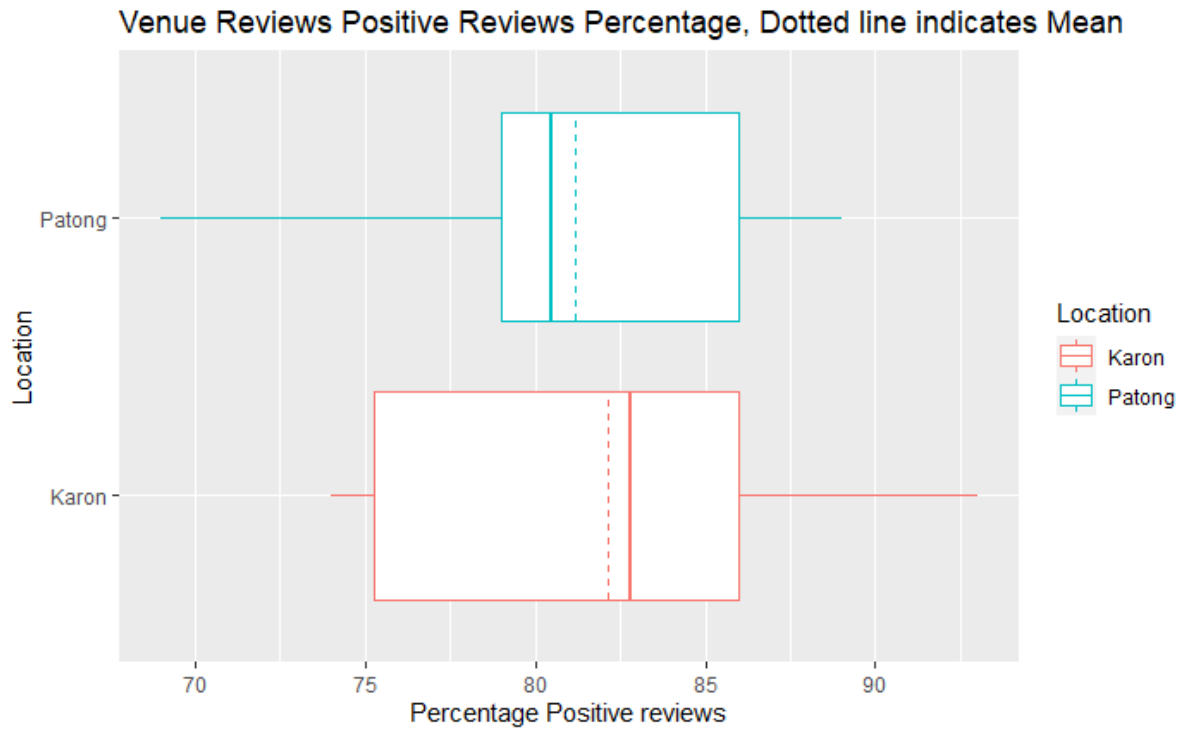


Figure 71 - Boxplot Positive Reviews %

It was noted that the mean positive percentage for both locations decreased to 81.2% for Patong and 82.6% for Karon. This implies that the hypothesis of several positive words in a positive review skewing the result may be true. Karon maintained the left skew observed when analysing words but it was significantly reduced in magnitude. Patong also developed a slight right skew. Patong now has the larger range of values at 20%, Karon has a range of 19%. Despite the slight differences in range, Patong has a significantly smaller IQR, again implying greater consistency in venues. However, Patong has a much smaller minimum value and Karon has a much higher value: The worst and best venues are in Patong and Karon respectively.

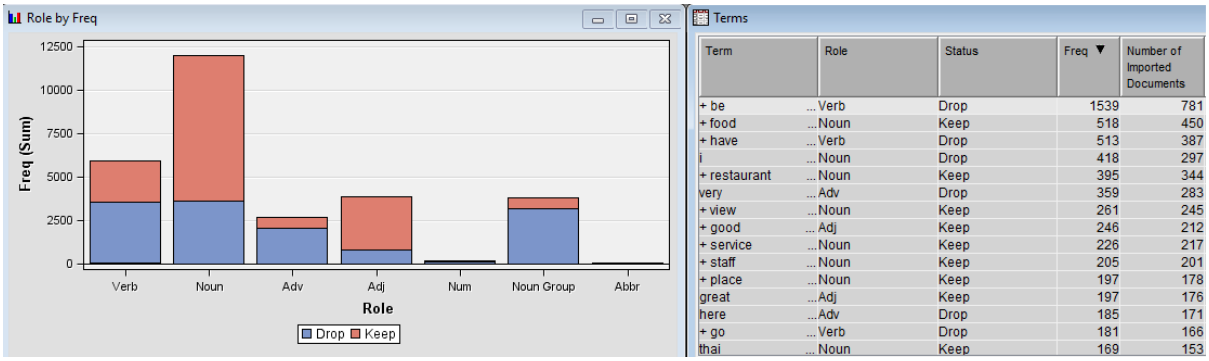
In summary, when measured by positive words and positive reviews, Patong has a greater consistency of quality of Hotels and restaurants and would therefore be the destination for a conservative traveller, even if there is a slight risk of attending the worst venue analysed. If one is searching for only the very best Restaurants and Hotels, and is willing to take a slightly greater risk, Karon should be their destination.

Implementation in SAS

Using the two CSV files written for each location in the data preparation steps, the reviews were imported using 2 separate file import (FI) nodes. Initially there was significant difficulty in importing all 1000 reviews. The Patong CSV file was only yielding 134 rows, the Karon CSV file was yielding 419. Not only were there rows missing, the rows that were successfully imported contained new columns (VAR1, VAR2), which contained a mixture of ID numbers and review texts. The CSV files were

manually scanned in Microsoft Excel and it was confirmed that both contained 1001 rows, including the headers, all in appropriate columns. The CSV was then opened in Microsoft Notepad. This revealed that the line breaks ("\n") that were removed in the R implementation, remained in the CSV files. SAS was reading the line breaks as the beginning of a new column, which was disrupting the order of the columns. This is because they were written to CSV before the further pre-processing steps were carried out. To correct this oversight the CSV files were deleted, and new CSV files written after the tm_map pre-processing functions but before stemming. This meant the new CSV files contained only two features: an index number and the review text itself. The review text was all lower case, with punctuation, hyperlinks, special characters, and line breaks removed. Using the FI node the new CSV files were imported and the issues with new column creation had been resolved, however there were still only 134 and 419 rows, respectively. The rows where the import had ceased were investigated in Microsoft Excel. These rows were found to contain double spaces which SAS was interpreting as the end of the data and was ceasing the import. These instances of double spacing were removed from both CSV files and FI was re-run. Once satisfied the data had been successfully imported, the variable roles were set. The ID column was set to the 'ID' role and the review column was set to the 'Text' Role.

Next a 'Text Parsing' (TP) node was used to parse the text. This essentially breaks down the text in the reviews into its component parts so that the reviews meaning may be understood. The TP node achieves this by ignoring parts of text such as conjunctions, pronouns, numbers, punctuation, and prepositions. Even though this had already been completed in Rstudtio, it was repeated in case something was missed. As the review column was set to the 'text' role, the parsing node automatically knows which column to parse. A 'Text Filter' (TF) node in conjunction with an 'English Dictionary' file provided by the University of Salford. The TF node filters the remaining word types and keeps ones that may be useful for text analysis and drops those that would not be useful.



Term	Role	Status	Freq ▼	Number of Imported Documents
+ be	Verb	Drop	1539	781
+ food	Noun	Keep	518	450
+ have	Verb	Drop	513	387
i	Noun	Drop	418	297
+ restaurant	Noun	Keep	395	344
very	Adv	Drop	359	283
+ view	Noun	Keep	261	245
+ good	Adj	Keep	246	212
+ service	Noun	Keep	226	217
+ staff	Noun	Keep	205	201
+ place	Noun	Keep	197	178
great	Adj	Keep	197	176
here	Adv	Drop	185	171
+ go	Verb	Drop	181	166
thai	Noun	Keep	169	153

Figure 72 - Word Role Frequency Plots

It was noted that a large proportion of nouns and adjectives were dropped by this node: 'Very' was dropped as the sentiment cannot be understood without a supporting adjective such as 'very good'. Dropping 'very' will leave the supporting adjectives that may be understood independently: 'Very good' becomes 'good'. While the TF node generally filtered the words appropriately, the 'filter viewer' function was used to override any relevant words that may have been overlooked. Several words were not dropped when they held no relevance to the analysis. For example, 'Ive', 'week' and 'know' were retained by the TF node but later manually dropped.

Finally, a 'Text Topic'(TT) node and a 'Text Cluster' (TC) were attached to the text filter node. The TT node analyses each word in the data, words frequently occurring together are clustered into topics.

The TC node clusters text in to groups and shows the descriptive terms for those groups. The TC node uses the term-document frequency matrix and transforms into a weighted, n-dimensional representation to cluster the words (SAS citation). The TC node was implemented using the expectation maximisation algorithm with low single value decomposition resolution. The TT node was configured to 'learn' 10 multi term topics, 10 was selected to see if there would be distinct topics for each venue in each location. Similarly, the text cluster node was configured to display a maximum of 10 clusters with a maximum of 10 descriptive terms per cluster.

Results in SAS

Firstly, the TF nodes' 'filter viewer' function was used to inspect the notable terms with high frequency, including some highlighted in the R implementation. The word 'View' has a high frequency in the Patong reviews which is consistent with the R results. Using the 'view concept links' function customers' opinion of the view is positive. Another term that was ambiguous in the R implementation was 'Price' within the Karon reviews. Viewing the concept links for this term most links indicate a positive opinion of the price, with one link indicating a negative opinion: 'high' price. The thickness of the links indicates the strength of association, in both cases the links to positive

terms are stronger.

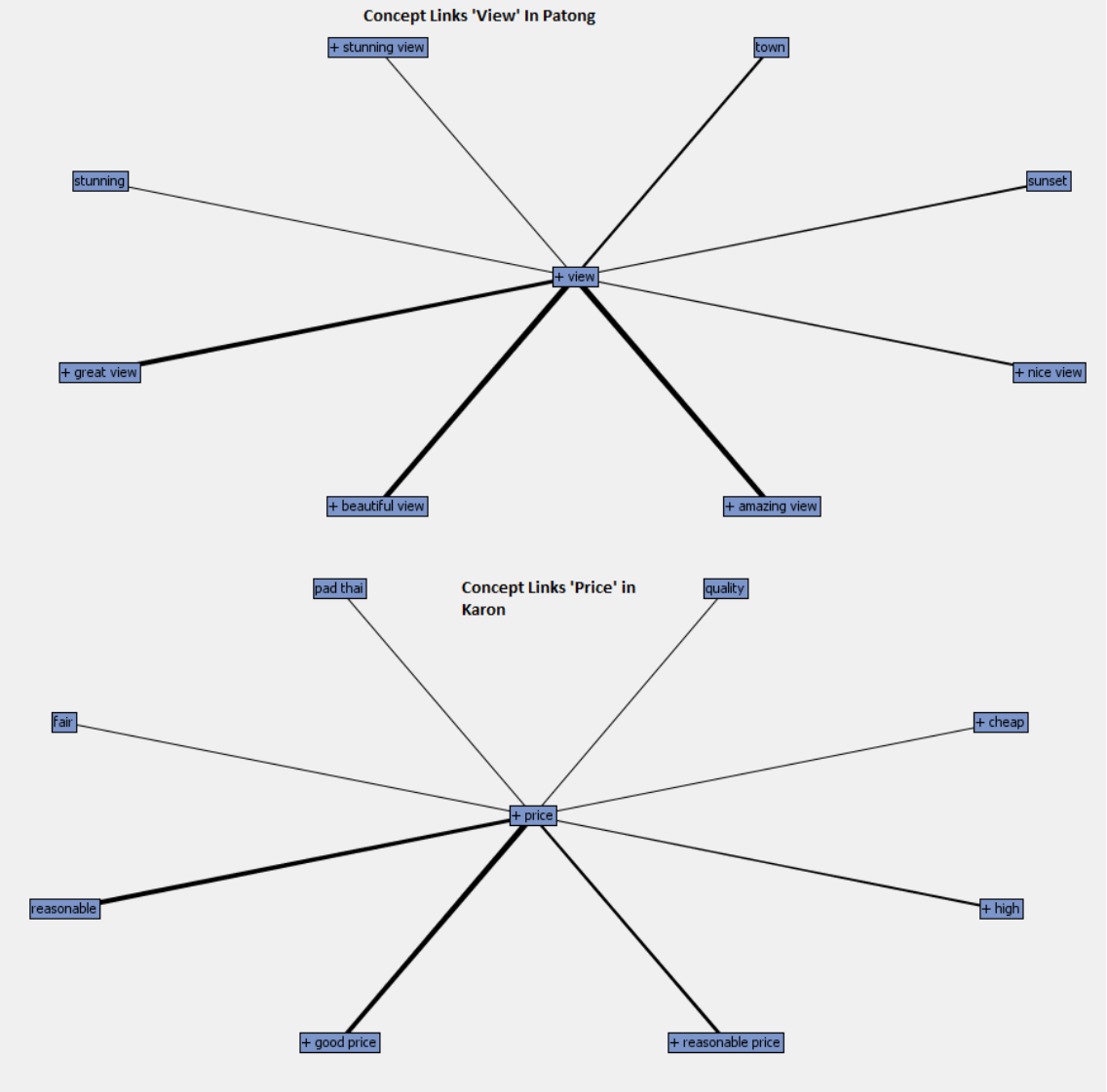


Figure 73 - SAS Link Diagrams

As ‘Food’ is one of the highest frequency words in both locations, and a potentially important one to potential customers, the concept links were inspected for both locations. Both locations had fully positive associations with ‘food’. Both had strong links with ‘Thai’ and ‘thai food’ indicating that a customer may get good quality Thai cuisine in both locations. It was noted that Karon also had associations with ‘Indian’, ‘Indian food’ and ‘Pizza’ indicating one may be able to get a wider range of cuisines in Karon. As only 10 venues from each location were analysed, this can not be asserted with confidence as this could be due to sampling error.

Next the TT nodes results were inspected. Comparing the topics from both locations there are 7 distinct cuisines mentioned in the topics generated from Karon and only 3 in Patong. This supports what was found within the ‘concept links’: That Karon offers a greater diversity of food venues.

Patong Topics

Topic ID	Topic	Number of Terms	# Docs
1	gritta,la gritta,amari,+stay,+hotel	52	71
2	german,thai,+breakfast,+buffet,+good	76	128
3	phuket,+hill,town,phuket town,+view	60	71
4	italian,patong,best,+good,good	51	93
5	+time,always,+place,+yacht,chalong	81	134
6	+road,bangla,+find,patong,tuk	68	110
7	+table,+book,birthday,+dinner,+arrive	83	129
8	thai,+thai food,+recommend,+service,+food	62	144
9	+staff,+friendly,+nice,+friendly staff,great	80	120
10	+order,+dish,+dinner,+salad,+curry	83	133

Karon Topics

Topic ID ▲	Topic	Number of Terms	# Docs
1	cuisine,palm cuisine,palm,+stay,+hotel	60	74
2	garlic,+bread,garlic bread,+fruit,free	40	59
3	+staff,+friendly,great,+meal,+price	63	160
4	+pizza,+eat,italian,+pasta,+night	51	121
5	indian,+indian,+owner,gaurav,+taste	60	88
6	+trip,+time,+review,+visit,karon	70	141
7	+curry,rice,+order,fried,chicken	77	106
8	thai,+thai food,+food,karon,+beach	57	144
9	+healthy,raw,+place,+vegan,+salad	67	106
10	+steak,+order,+menu,el,gaucho	83	128

Figure 74 - Food Topics

Finally inspecting the TC node results further corroborated the findings within the other nodes.

Patong Clusters

Cluster ID ▲	Descriptive Terms	Frequency	Percentage
1	+staff +friendly +recommend highly definitely delicious +place excellent +nice great ...	177	18%
2	german +wine +beer +breakfast +price +menu +good quality +eat kamala ...	117	12%
3	+order +curry chicken +dish +dinner great little good +meal main ...	180	18%
4	thai +'thai food' town 'phuket town' +hill phuket +ring +food +look +view ...	89	9%
5	italian gritta 'la gritta' amari advisor 'dolce vita' dolce vita 'trip advisor' trip ...	89	9%
6	+road bangla 'bangla road' tuk 'tuk tuk' main +find worth little top ...	45	5%
7	+table +book +hotel always +time +dine +visit patong sunset +view ...	150	15%
8	birthday sea wonderful amazing beautiful perfect +view sunset +service fantastic ...	98	10%
9	+buffet +schnitzel pork kamala +salad kokosnuss german chicken +owner +time ...	55	6%

Karon Clusters

Cluster ID	Descriptive Terms	Frequency	Percentage
1	+healthy raw +vegan +happy +farm +salad +place +love fresh +drink ...	67	7%
2	+meat el +steak +quality +dinner +menu +night +drink +salad didnt ...	138	14%
3	chicken +curry rice fried green lunch +order +salad +dish +farm ...	74	7%
4	+pizza italian +pasta da +cheese didnt +order +steak +night +good ...	72	7%
5	+recommend highly +beach +good karon great delicious excellent +servi...	210	21%
6	indian +indian gaurav tandoori +flame 'indian food' +'tandoori flame' +'indi...	49	5%
7	garlic +bread +stay 'garlic bread' +fruit water +hotel free +review +dinner ...	103	10%
8	+staff +friendly +'thai food' thai +price food +food great +service back ...	220	22%
9	+trip +day advisor 'trip advisor' +time +review first +visit +night didnt ...	67	7%

Figure 75 - Text Cluster Node Results

Both TC nodes were configured to produce a maximum of 10 clusters. Both nodes only produced 9. Once again, Karon shows a greater variety for cuisines and that Patong is perceived to have a nice view. It can be seen in Karon cluster 5 that 'Beach' is clustered with positive adjectives, implying Karon has a nice beach. That all the terms within the clusters and topics are positive indicates that the majority of sentiment in the reviews is positive: mostly positive reviews. If there was significant negative sentiment, it would likely appear in the clusters. The other explanation for lack of negative sentiment in the topics and clusters would be that the negative sentiment is spread over a broad range of topics. It can be concluded that most reviews show a positive opinion towards the food and accommodation in both locations. In summary the SAS results show that both locations have mostly positive opinions.

Results Comparison and Conclusion

The implementations in SAS and RStudio both achieved similar results but did so in different ways. In RStudio, the positive and negative reviews were broken down into ratios of positive and negative words and then classified into positive and negative reviews. This gave a quantitative assessment of the quality of venues in both locations. The term-document frequency plot allowed for interesting comparison of vocabulary most used in reviews in both locations. This quantitative analysis also yielded statistics that are of use when comparing locations, for example: When comparing the IQR of percentages of positive reviews, Patong has an IQR of 7 and Karon has an IQR of 11. A data scientist or statistician can immediately interpret this data as meaning Patong's venues are more consistent, but a lay person would not be able to interpret this. This however does not mean that these statistics are not of use. The statistics generated such as IQR, mean positive review percentage and range of positive review percentages could be used to derive a function for a weighted score for each location. However, a score function would be arbitrary without first surveying potential customers and ascertaining what features are important to them. This importance would be used to calculate the weights of the score function.

The SAS implementation contained quantitative elements and the results of the term-document matrix in the TF node were identical to the result in R. The SAS analysis allowed for more in-depth analysis of specific sentiment towards elements within each location: In R the 'View' in Patong was identified as noteworthy but it was unclear if the sentiment towards it was positive or negative. SAS was able to determine that sentiment towards the view was overwhelmingly positive. While R was able to generate a term-document frequency matrix, the topic and cluster analysis in SAS allowed for greater insight into sentiment towards different elements by not only displaying the frequency of individual words, but groups of words that were used frequently together. This gave context to the words being used and therefore greater understanding of the words intended meaning. These insights could be utilised to categorise locations: Locations good for dining out, locations with a good beach, locations with a specific cuisine.

SAS was able to extract sentiment with more context, this information could be presented to potential customers in the form of a dashboard, allowing customers to filter or search for terms that are important to them without much further manipulation. The information gained in R would require further processing for the general consumer to understand. In conclusion both platforms were able to mine information from the data that can be used to inform customers decision when choosing a location for a vacation.

REFERENCES

How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read. (2020). Retrieved 15 December 2020, from https://www.bernardmarr.com/default.asp?contentID=1438

Using the Text Cluster Node :: Getting Started with SAS(R) Text Miner 12.1

Liu, B., 2015. *Sentiment Analysis*. Cambridge University Press, pp.1-10.

Liu, B., 2015. *Sentiment Analysis*. Cambridge University Press, pp.17-68.

Wickham, H. 2016 *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Wickham, H. 2019*. stringr: Simple, Consistent Wrappers for Common String Operations. R package* version 1.4.0. https://CRAN.R-project.org/package=string

Wickham, H.François, R. Henry, L. and Müller, K 2020. *dplyr: A Grammar of Data Manipulation.* R package version 1.0.2. https://CRAN.R-project.org/package=dplyr

Fellows, I. 2018. *wordcloud: Word Clouds*. R package version 2.6. https://CRAN.R-project.org/package=wordcloud

Müller, K and Wickham, H (2020*). tibble: Simple Data Frames.* R package version 3.0.4. https://CRAN.R-project.org/package=tibble

Jeffrey B. 2019. *ggthemes: Extra Themes, Scales and Geoms for 'ggplot2'.* R package version 4.2.0. https://CRAN.R-project.org/package=ggthemes

APPENDIX

Sentiment Analysis Code Thomas Madeley.pdf

Sentiment Analysis Thomas Madeley.Rmd

CLICK PDF FILE TO VIEW CODE.