Contents

1 Question 1

A transnational dataset is used, with all the transactions from 01/12/2010 to 09/12/2011, for a UK-based and registered non-store online retail. The company mainly sells unique gifts for all occasions. Most customers are wholesalers. The raw dataset contains 541,909 instances and 8 attributes. Data pre-processing is performed and the exploratory data analysis is used to identify key findings. Market basket analysis (MBA) is conducted, followed by conclusions and recommendations.

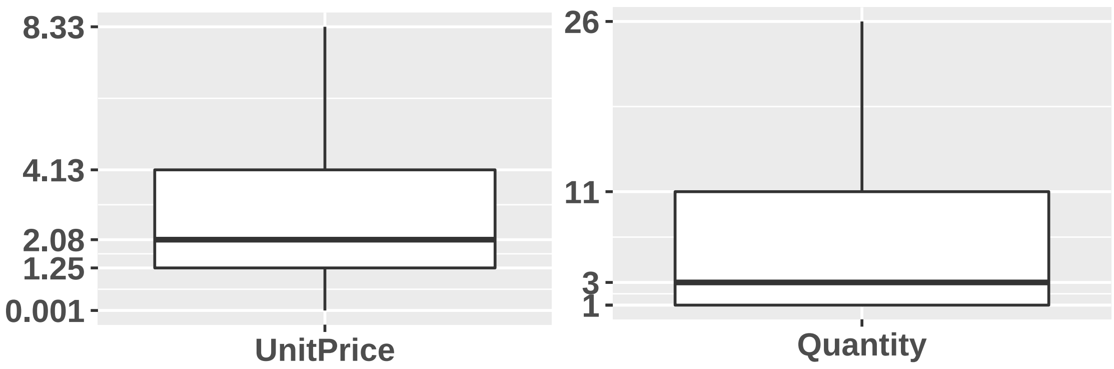
1.1 Data Preprocessing

The `read_excel` function from `readxl` package is used to read xlsx file and detect automatically the forms of data. We specify data types to accelerate large data importing process and avoid warnings, where `POSIXct` represents date-time column. Data cleaning is vital to the study. Missing values are identified, especially in `Description` (count: 1,454) and `CustomerID` (count: 135,080). Missing values in `Descrption` are dropped as association rules learning is focused on transactions and missing descriptions are useless. However, missing values in `CustomerID` are retained as they do not affect the learning process. Negative quantities are found in `Quantity` probably due to refunds. Thus, we only keep orders of positive quantity. `UnitPrice` contains negative and 0 unit prices, probably resulted from items missing, disposed, stolen, etc., which are removed. The resulting dataset contains 530,104 instances.

Outliers are found in `Quantity` distribution. Though most quantity values are ≤ 10 with an average of 10.54, there is a wide range (1~80,995) with 132,247 instances larger than 10 and 107 instances even greater than 1,000. This may be led by transactions of wholesalers. For `UnitPrice` distribution, there is also a large spread (£0.001~£13,541.33). Most prices per product are $\leq$ £4.130, with an average of £3.908. Price outliers are investigated to find odd descriptions. Possible interpretations are provided, such as `POSTAGE` (spending on mail services), `Adjust bad debt` (related to budgeting), `Manual` (installation services), etc. The associated undesirable descriptions are identified and removed using these words. Consequently, the dataset contains 527,947 instances, with 13,962 instances removed in total.

1.2 Exploratory Analysis

By calculating the statistics of boxplot, outliers for `UnitPrice` and `Quantity` are removed to reduce skewness. Next, boxplots are plotted with five statistics as below: `lower` (25% quantile), `middle` (median), `upper` (75% quantile), `ymin` (smallest observation $\geq$ `lower` - 1.5 * IQR, where IQR is the distance between the first and third quartiles) and `ymax` (largest observation $\leq$ `upper` + 1.5 * IQR). From Figure 1, the median of unit price is £2.08 while the median of quantity is 3 with extreme values excluded.



(a) Distribution of unit prices **(b)** Distribution of quantities

Figure 1: Box plots for UnitPrice and Quantity distributions (outliers removed).

Aggregated by `Description`, the top 20 descriptions reordered by the number of transactions are plotted in Figure 2. The most purchased items is white hanging heart t-light holder. Products purchased show a preference for food and decorations. Top 10 countries by the number of transactions are plotted in Figure 3, excluding the UK due to the largest number of transactions (484,082). With the UK excluded, the top 8 countries with the largest number of transactions are all from Europe, among which the top 3 are Germany, France and EIRE with each greater than 7,500.

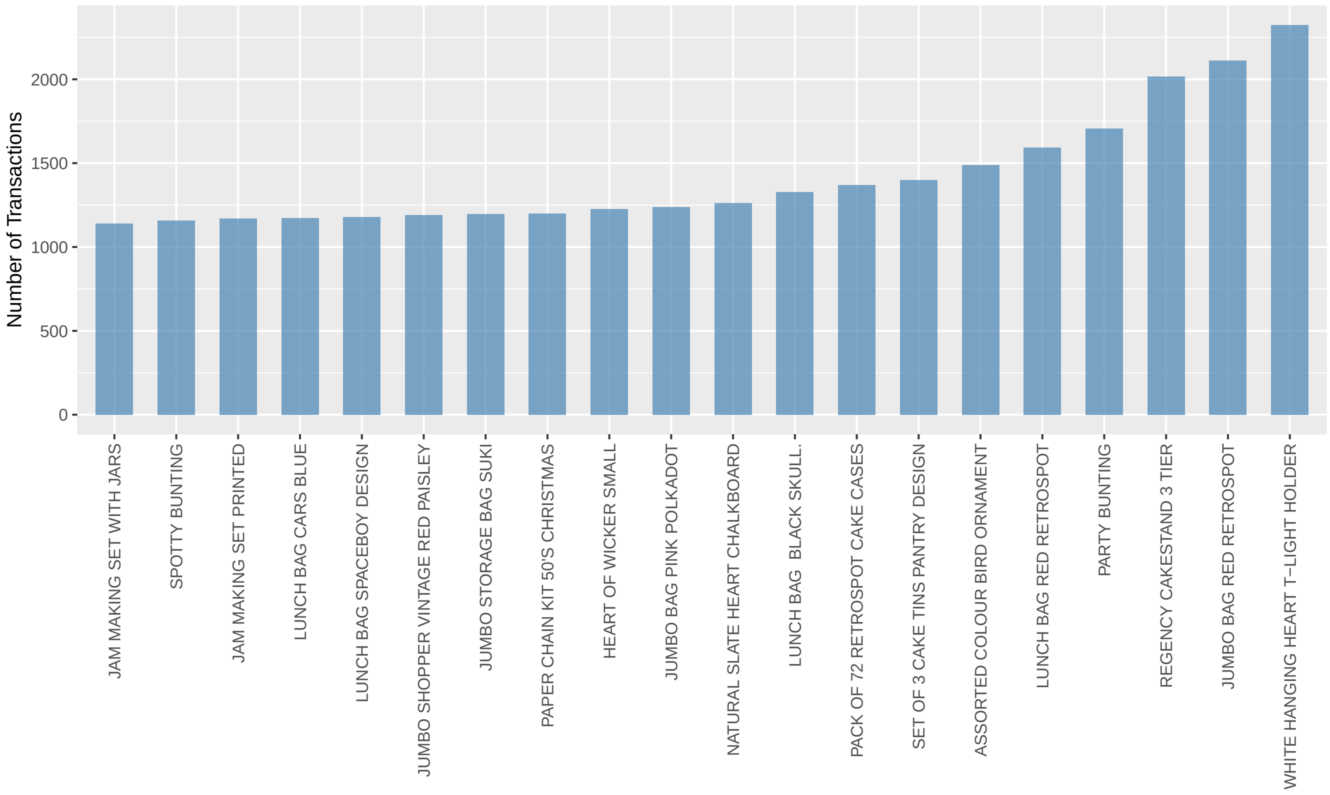


Figure 2: Top 20 most purchased items ranked by the number of transactions.

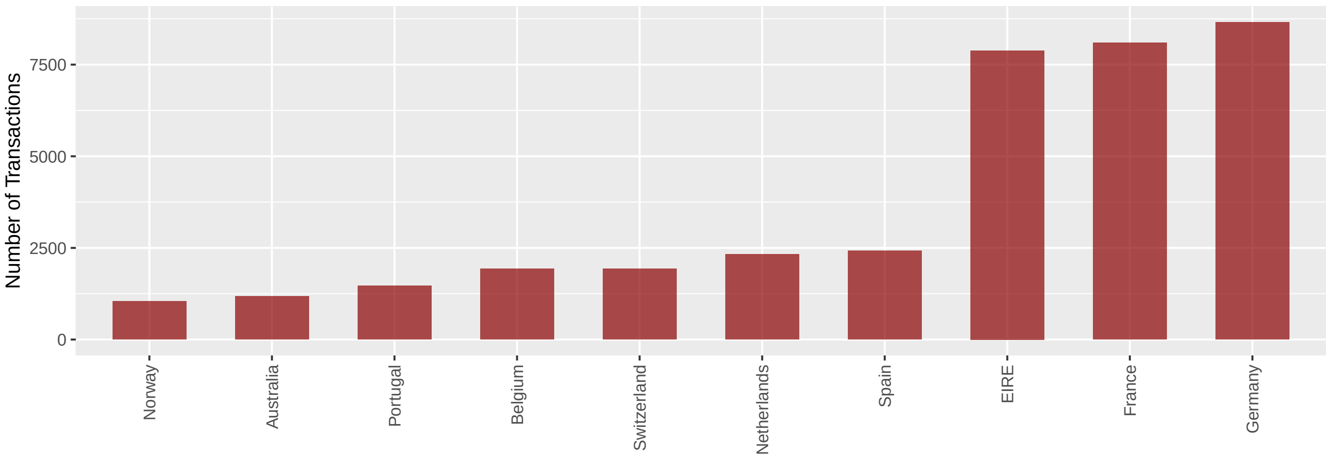


Figure 3: Top 10 countries that have the largest number of orders (UK excluded).

The `InvoiceDate` column is split into `Date` and `Time` to extract year-month-day and hour-minute-second information from original transaction data and aggregated by months, week days and hours. Monthly number of transactions (12/2010~12/2011) is shown in Figure 4. An increasing trend is observed. 11/2011 shows the largest number of transactions of 83,117, probably due to holiday purchase for gifts. The smallest is in 12/2011, due to insufficient data recorded only till 09/12/2011. In Figure 5, daily number of transactions is counted by the total number of transactions throughout the whole week. The largest number of transactions is 100,601 on Thursday, while the smallest number of transactions is 63,785 on Sunday, excluding Saturday with no transaction recorded. Perhaps, Saturday is a break of order-placing. Similarly, hourly number of transactions is counted by the total number of transactions throughout the whole day. Peak hours (12PM~15PM, UK time) have the number of transactions over 60,000 per hour, while scarse transactions happen around midnight, seemingly indicating international/non-European transactions.

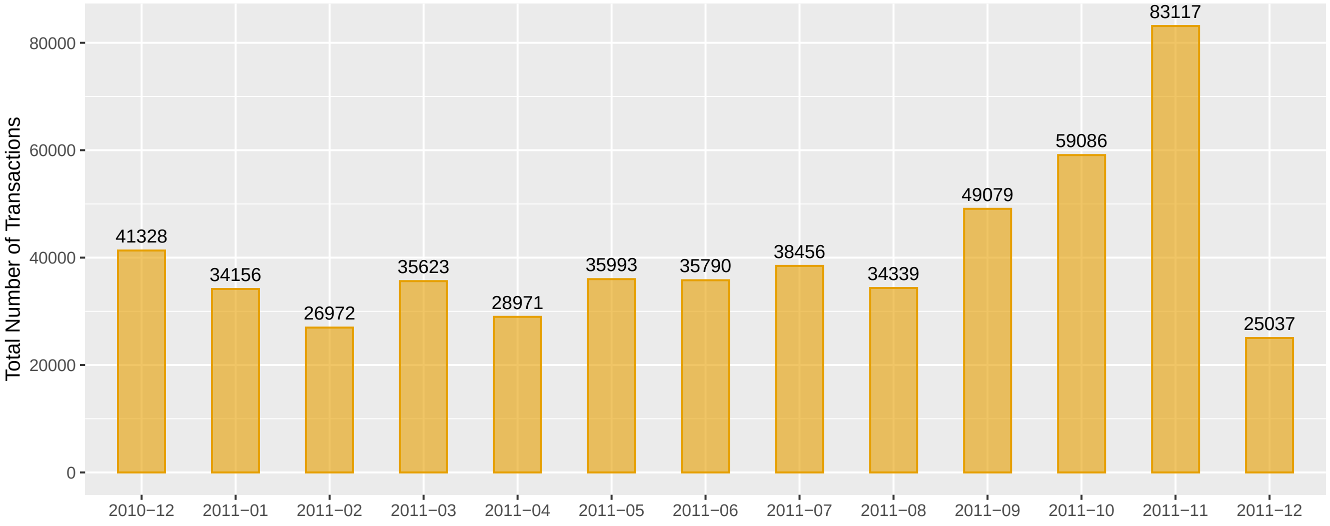
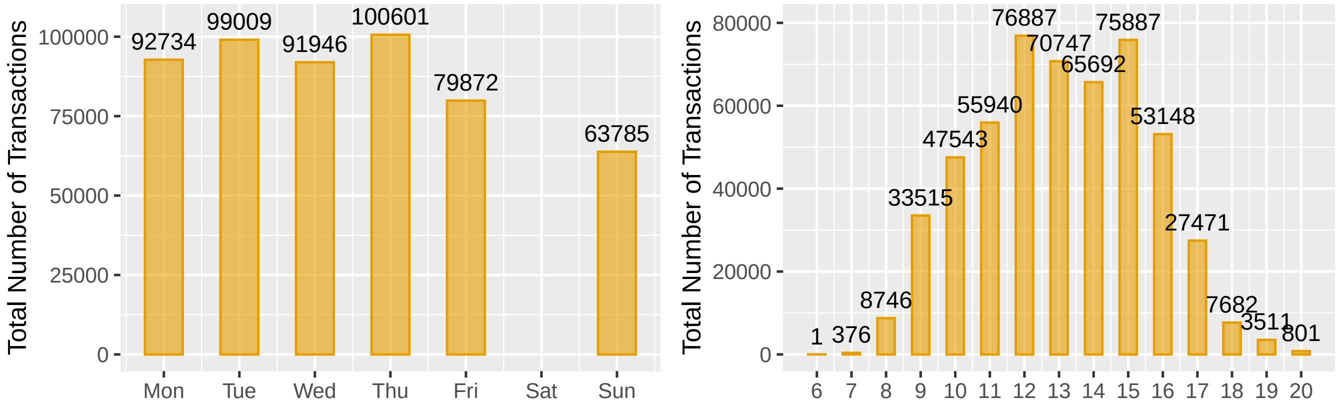


Figure 4: Visualisation of the number of transactions by months.



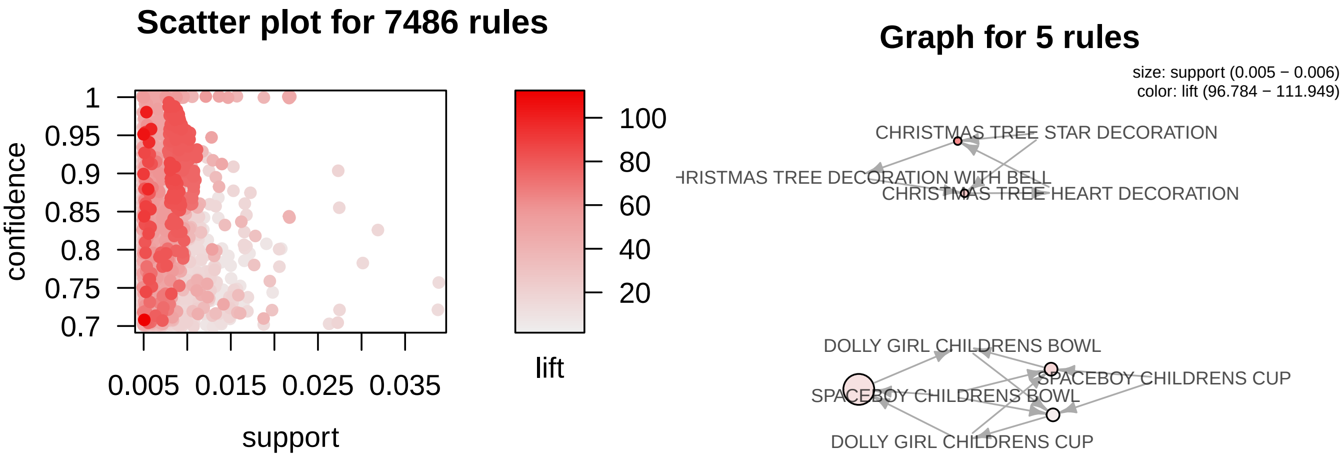
(a) In a week

(b) In a day

Figure 5: Total number of orders throughout a week or a day.

1.3 Market Basket Analysis

MBA is used to understand associations between products and identify rules by searching for groupings of products purchased together recurrently in transactions. The `transactions` object has 19,789 rows as itemsets with the smallest length of 1 (only 1 item in the transaction) and the largest length of 1,115. The Aapriori algorithm is used for mining recurring itemsets and identifying association rules. Parameters are specified as: `sup` (how often the rule occurs), `conf` (how reliable the rule is), `minlen` and `maxlen` (the smallest and largest number of items involved). Different parameters are used to set 2 user-defined criteria (minimum support and confidence) and different rule lengths.



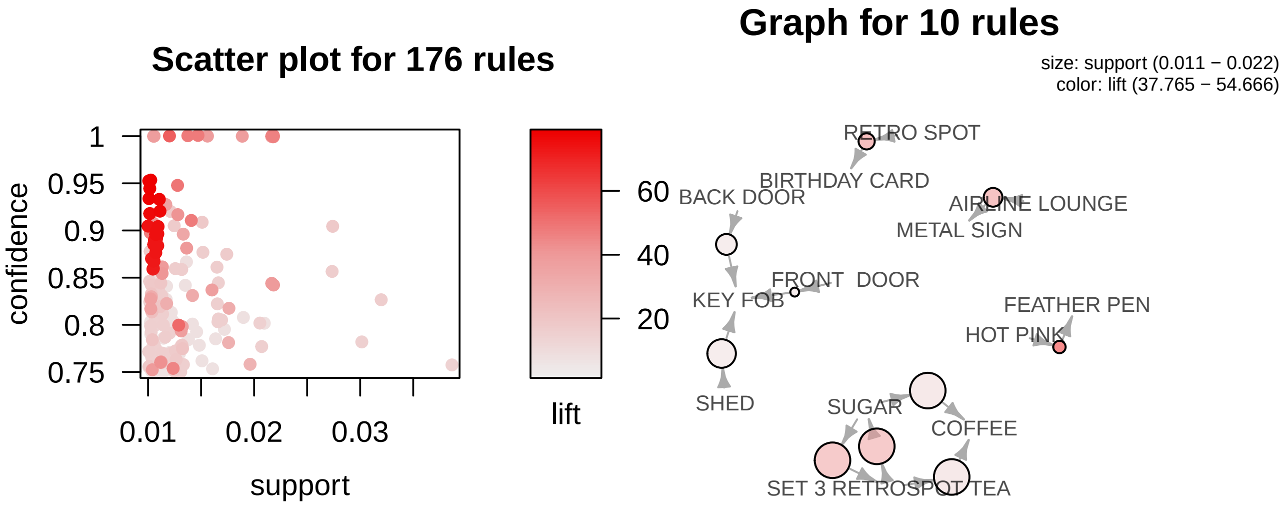
(a) Rules with attributes

(b) Top 5 rules of highest lift values

Figure 6: Association rules for sup.th = 0.005, conf.th = 0.7, involving 2 to 10 items.

Association rule 1: A total of 7486 association rules are found, with support at least 0.5% and confidence at least 70%, involving 2 to 10 items. In Figure 6, darker red shows high lift, indicating high support for a joint rule relative to the multiplied support of its parts taken separately. Most rules are of small support, considering the large number of itemsets (19,789), where 0.5% represents around 99 itemsets. From the top 5 rules with highest lift, it shows that Christmas tree ornaments are more likely to be bought together as seasonal purchase combinations, where the rules do not occur often as indicated by support values.

Association rule 2: A total of 176 association rules are detected with support at least 1% and confidence at least 75%, involving no fewer than 2 items. Figure 7 shows the top 10 above rules with highest confidence, suggesting that, for example, sugar and tea lead to purchases of coffee more often.



(a) Rules with attributes **(b)** Top 10 rules of highest lift values

Figure 7: Association rules for sup.th = 0.01, conf.th = 0.75, involving at least 2 items.

To understand what customers bought before buying a particular item or what they might buy if purchasing an item, the antecedent (item on the left of the rule) or consequent (item on the right of the rule) can be specified. The following rules are inspected and reordered by lift or confidence in Figure 8 and 9.

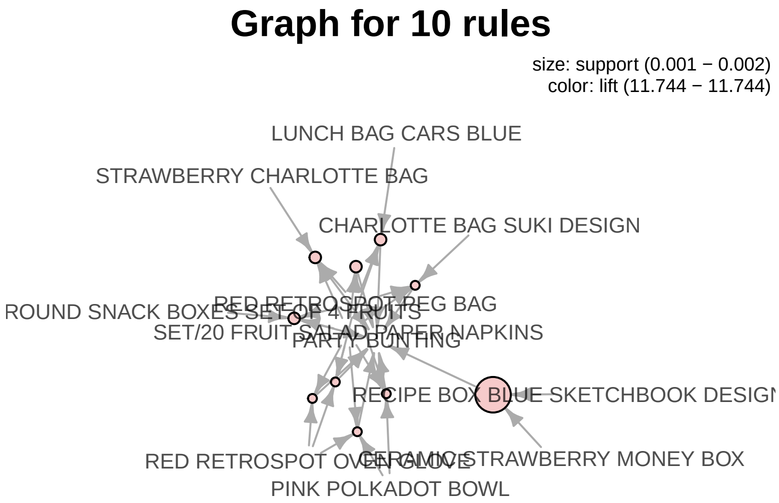
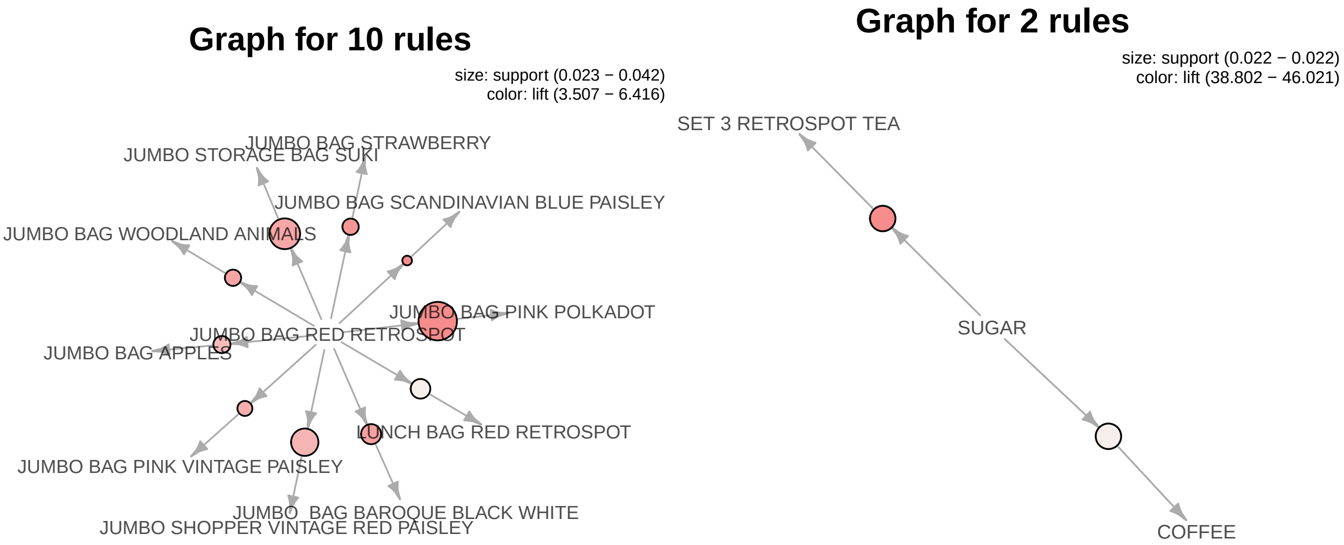

Graph for 10 rules
size: support (0.001 − 0.002)
color: lift (11.744 − 11.744)
LUNCH BAG CARS BLUE
STRAWBERRY CHARLOTTE BAG
CHARLOTTE BAG SUKI DESIGN
RED RETROSPOT PEG BAG
ROUND SNACK BOXES SET OF 4 FRUITS
SET/20 FRUIT SALAD PAPER NAPKINS
PARTY BUNTING
RECIPE BOX BLUE SKETCHBOOK DESIGN
RED RETROSPOT OVEN GLOVE
CERAMIC STRAWBERRY MONEY BOX
PINK POLKADOT BOWL

Figure 8: Association rules of consequent = PARTY BUNTING.


Graph for 10 rules
size: support (0.023 − 0.042)
color: lift (3.507 − 6.416)
JUMBO BAG STRAWBERRY
JUMBO STORAGE BAG SUKI
JUMBO BAG SCANDINAVIAN BLUE PAISLEY
JUMBO BAG WOODLAND ANIMALS
JUMBO BAG PINK POLKADOT
JUMBO BAG RED RETROSPOT
JUMBO BAG APPLES
LUNCH BAG RED RETROSPOT
JUMBO BAG PINK VINTAGE PAISLEY
JUMBO BAG BAROQUE BLACK WHITE
JUMBO SHOPPER VINTAGE RED PAISLEY
Graph for 2 rules
size: support (0.022 − 0.022)
color: lift (38.802 − 46.021)
SET 3 RETROSPOT TEA
SUGAR
COFFEE

(a) Antecedent = JUMBO BAG RED RETROSPOT

(b) Antecedent = SUGAR

Figure 9: Association rules of specifying antecedent.

Association rule 3: The aim is to identify products purchased before buying party bunting. There are 16,387 rules with support at least 0.1% and confidence at least 90%. Among the top 10 lift rules, customers are likely to buy fruit, salad, paper napkins and red peg bags before party bunting.

Association rule 4: To identify potential purchases when buying jumbo red bag first, 67 rules are found with support at least 0.1% and confidence at least 10%. It seems that those are purchased by wholesalers who bought jumbo bags of different patterns and colours. The reason that the confidence values are low may be associated with the fact that the options of bag patterns and colours are rich.

Association rule 5: There are 2 rules identified for potential purchases along with buying sugar, with support at least 1% and confidence at least 70%. It is evident that sugar is often purchased first by coffee and tea drinkers, and tea is more likely to be purchased with a higher lift value of 46.02.

In conclusion, this company has a strong market presence in the UK and other European countries, with different types of gift-giving products for all occasions, particular to wholesalers for party decorations, jumbo bags, food and lunch bags. The company generates massive sales particularly during autumn and winter holidays. To expand in the long term and target international market, the introduction of gifts based on other festivals and occasions would be recommended. Incorporating colour schemes in similar products may increase the probability of buying products together. To promote sales of less popular combinations, coupons or discounts are recommended. The company should focus on how to grow international sales and target specific products to numerous customer segments, putting itself in a more competitive position.

2 Question 2

This study involves an exploratory analysis and the use of 4 classifiers, Random Forest, Boosting, Neural Network and kNN, on the `US-Stock.cvs` data to identify any key findings. This is executed by preprocessing the data and conducting the binary classification. Finally, a sectoral analysis is applied on financial services to compare the performance of classifiers.

2.1 Exploratory Analysis & Data Preprocessing

Among all the 4392 observations, 1346 belong to `class 0`, where stock price variation is negative, while 3046 are in `class 1`, where the price change is positive. Hence, around 69% of the stocks have a positive price variation. Figure 10 displays the `Class` distribution across all the 11 sectors, with each having more stocks in `class 1`. There is an imbalance ratio of 2.263 in the whole data, while financial services sector has the largest imbalance ratio, with over 700 stocks in `class 1` and around 100 stocks in `class 0`.

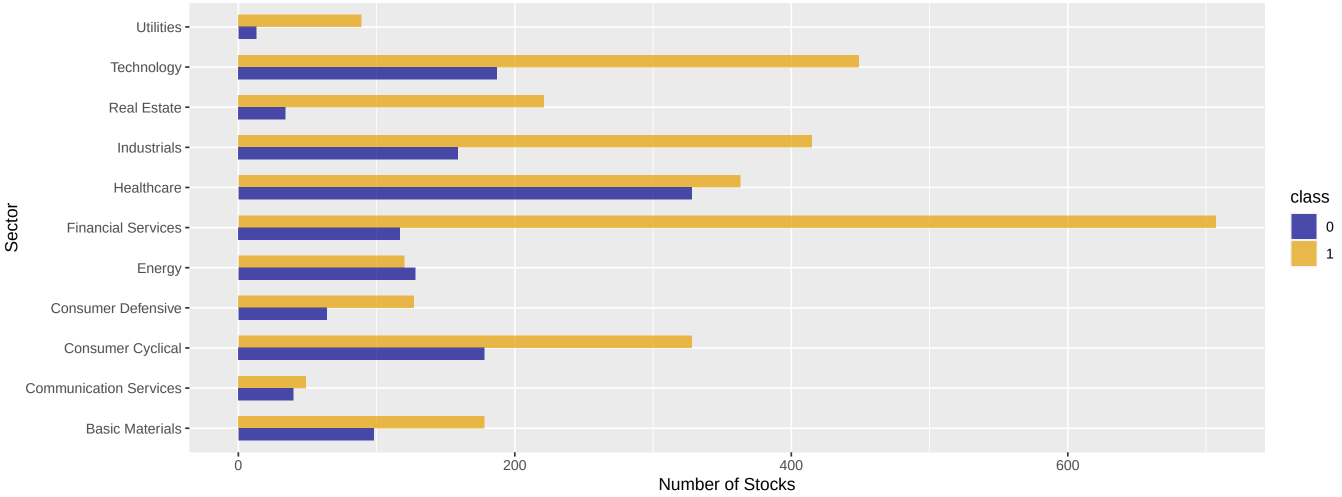


Figure 10: Class distribution across all the 11 sectors.

The percentages of missing values and zero values are calculated for all the 221 financial indicators. Figure 11 and 12 show the percentages in decreasing orders with the top 100 indicators in each plot. In particular, `cashConversionCycle` and `operatingCycle` have the highest proportions of missing values at nearly 100%. `Net.Income.Discontinued.ops`, `Preffered.Dividends` and `Deposit.Liabilities` contain large percentages of zero values, with all above 75%. The thresholds for NAs and zeros are computed using the quantiles of the percentage distributions. We decide to drop the top 25% NA-dominant variables, which are 54 indicators (with NA threshold at 10%), and the top 25% zero-dominant variables in the remaining, which are 43 indicators (with zero threshold at 24%). That being said, the variables with more than 10% of missing values or more than 24% of zero values among all the 4392 observations are removed, with the thresholds plotted in red dash line in Figure 11 and 12. Additionally, the variable `operatingProfitMargin` is deleted that has the same values for 2 classes. After handling missing, zero and singular values, the number of financial indicators is reduced to 123 from 221.

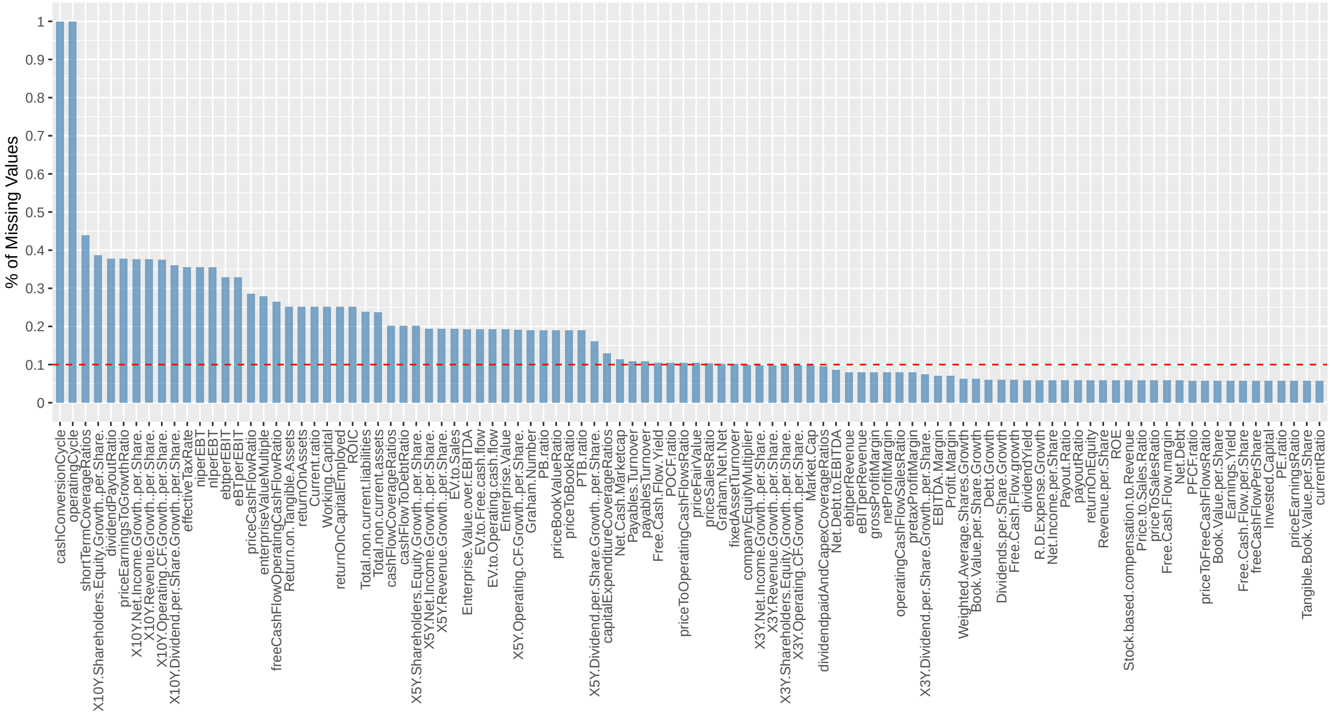


Figure 11: Top 100 indicators with the percentage of missing values.

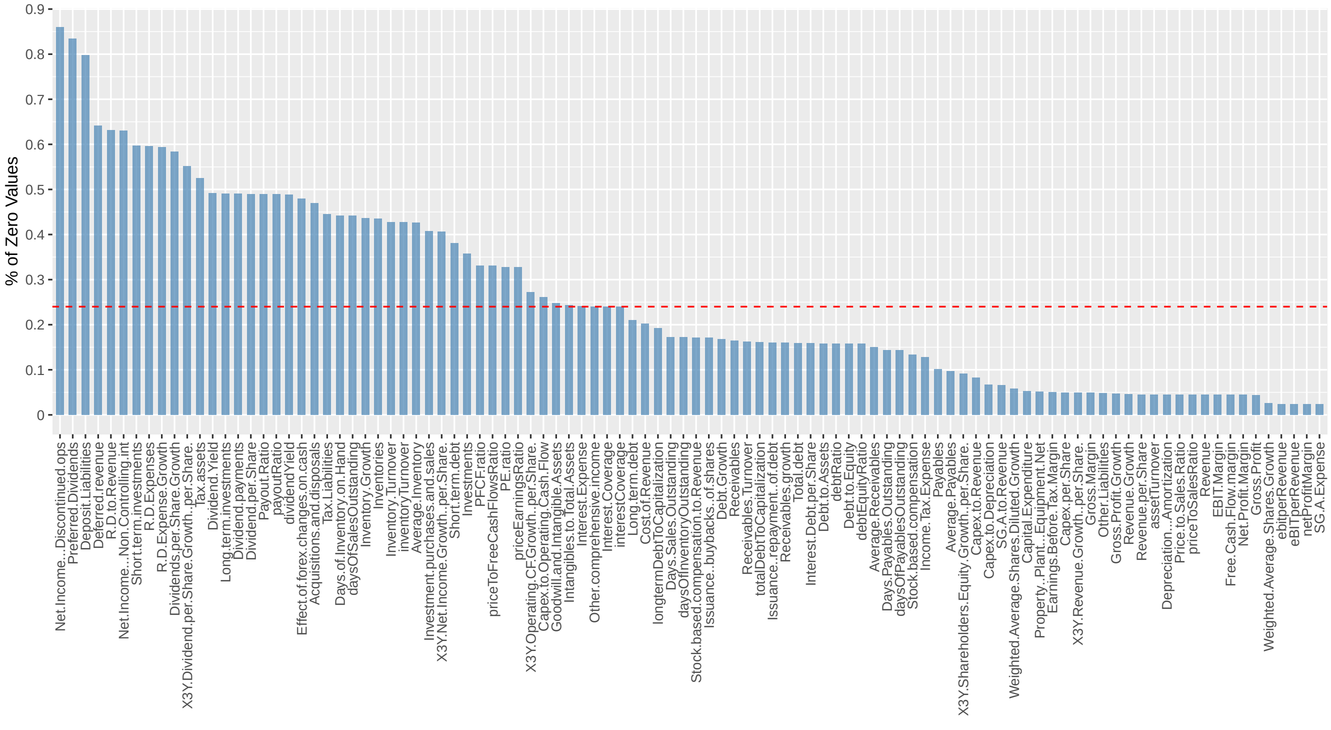


Figure 12: Top 100 indicators with the percentage of zero values.

To reduce skewness of each variable, outliers are replaced with values obtained through quantile computation. It is conducted by replacing zeros with NAs first and then computing statistics omitting both NAs and zeros. Specifically, we reuse the `boxplot.stats` function that is also applied in Question 1 to calculate `ymin` and `ymax`, which are lower and upper whisker of each variable. The extreme values (larger than `ymax` or smaller than `ymin`) are replaced with `ymax` and `ymin` values. Any missing or zero values are filled with the means aggregated by sector for each indicator, rather than filled with the mean of each column, to account for the difference of stock performances across sectors.

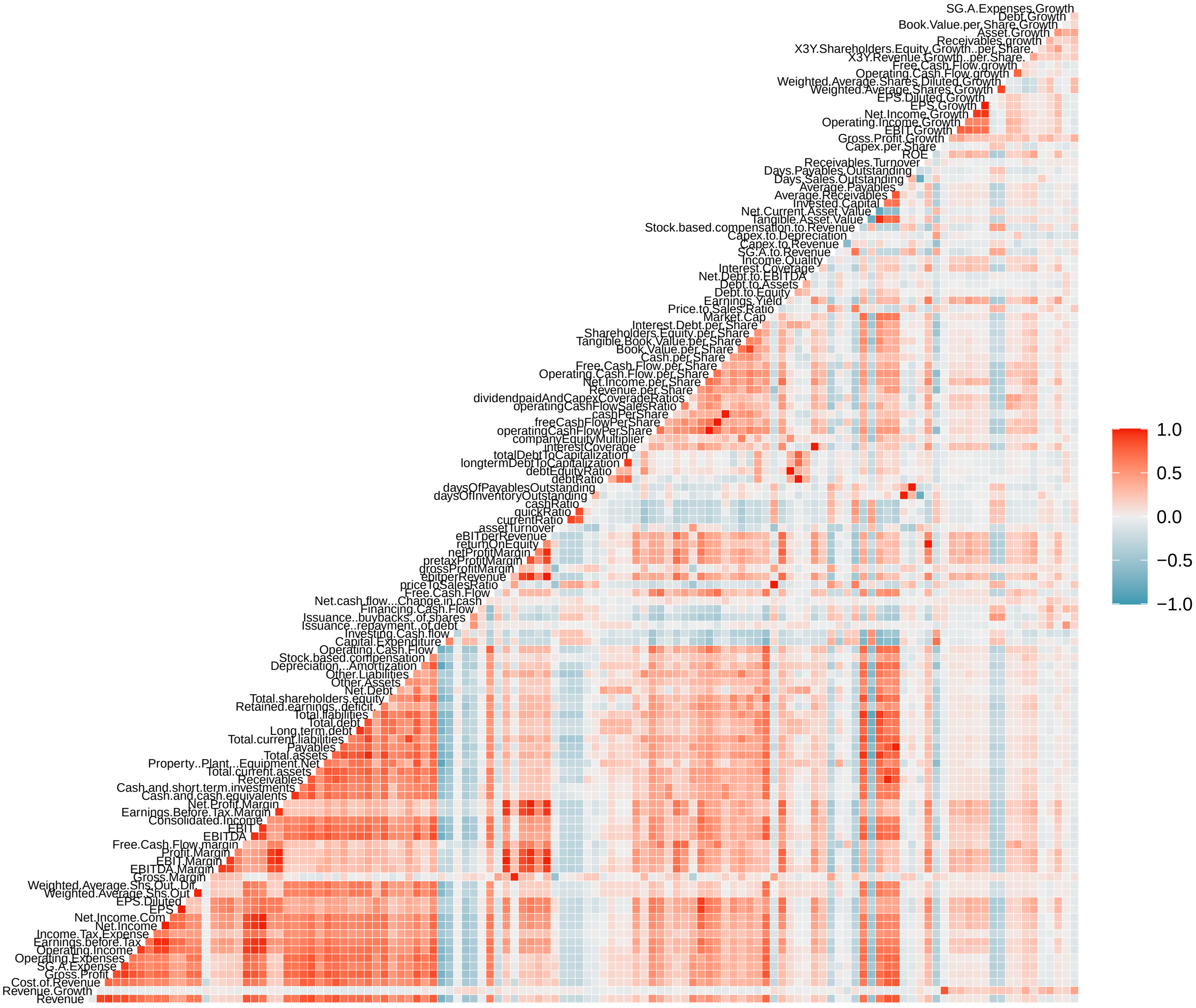


Figure 13: Correlation matrix of 123 financial indicators.

A correlation matrix is used to show the linear correlation between the selected 123 financial indicators. It is worth noting that the majority of variables are weakly or barely correlated before handling outliers and missing/zero values. In Figure 13, heavily correlated features are identified with dark red or blue.

For example, the highly positive correlations between `EPS.Diluted.Growth` and `EPS.Growth` and between `Weighted.Average.Shares.Diluted.Growth` and `Weighted.Average.Shares.Growth` make sense in reality, as these indicators provide similar financial information of a company.

2.2 Training & Comparison of Classification Models

In the subsequent analysis, Random Forest, Boosting, Neural Network and kNN methods are applied to conduct binary classification for all the 4392 observations. Prior to classification, the data is randomly split to the distinct training (70%) and test (30%) sets. The error rates and AUC values are calculated for various classification models on the test set to compare the classifer performance.

Random Forest

Since there are many highly correlated variables included in the data, the motivation of using Random Forest is to train less correlated trees by randomly selecting a number of variables as split candidates from the total 123 financial indicators. Using the `caret` package, the models are tuned via a 5-fold cross-validation on one parameter `mtry`, which refers to the number of variables randomly selected at each split. The tuning grid for `mtry` is specified as `mtry=c(2,11,30,50,70,90,110,123)`. Figure 14 demonstrates the relationship between cross-validated accuracy and `mtry`. The optimal `mtry` value is 110 with the highest accuracy of 0.724 and Kappa equal to 0.288. Additionally, the importance of variables is examined and it indicates that the accuracy decreases the most when excluding the indicator `Tangible.Asset.Value`, followed by `Price.to.Sales.Ratio` and `Earnings.Yield`.

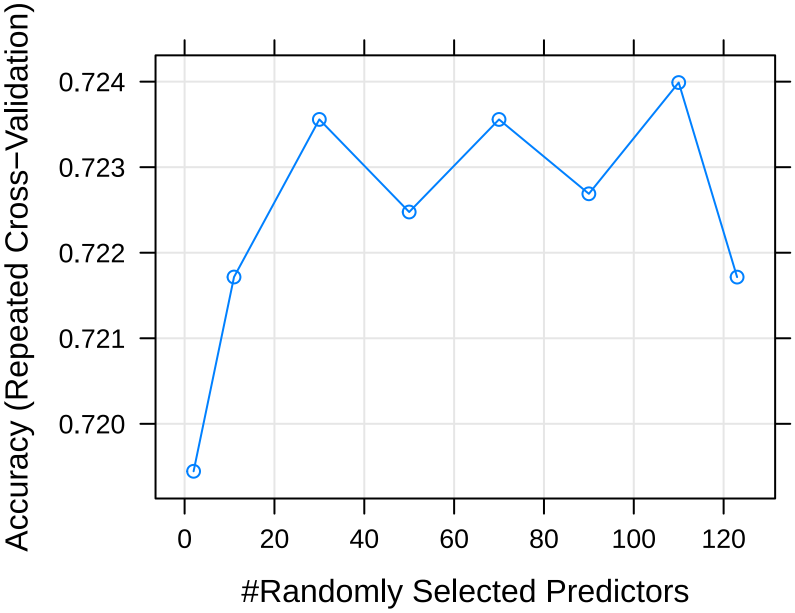
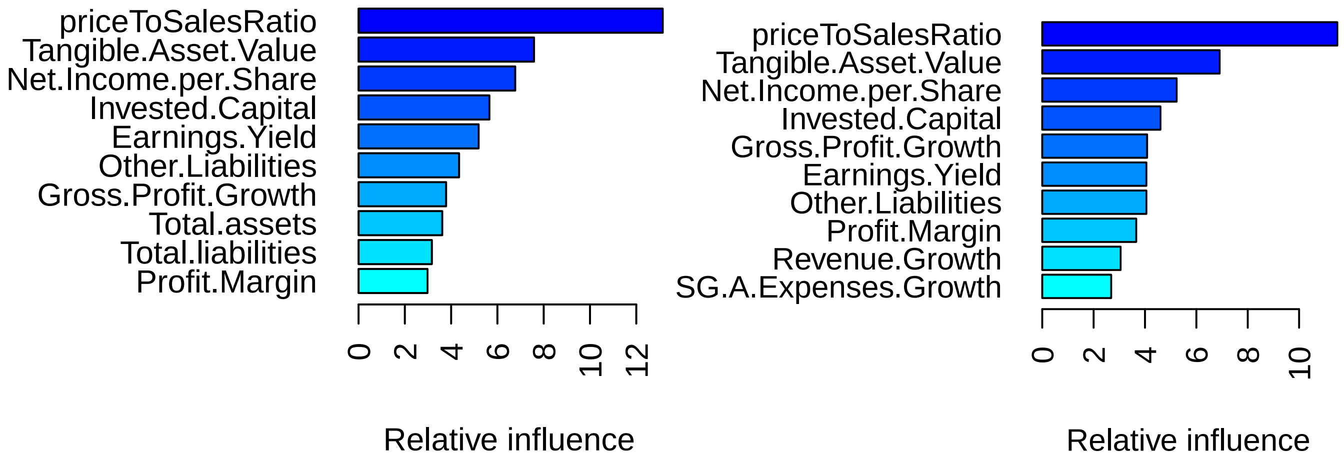


Figure 14: CV accuracy against mtry for Random Forest model.

Boosting

The motivation of using Boosting is to combine the results of many weak classifiers and produce a powerful learner. A Boosting model is first trained using Bernoulli distribution as loss function, and then compared with AdaBoost algorithm via `gbm` package. By using the out-of-bag (OOB) error to estimate the optimal number of Boosting iterations, the optimal `n.trees` for Gradient Boosting using the Bernoulli distribution and AdaBoost are 390 and 482 respectively. The relative influence of top 10 variables based on the chosen number of trees is visualised in Figure 15. From both Boosting methods, `priceToSalesRatio` has the highest relative influence, accounting for more than 10% of the reduction to loss function, followed by `Tangible.Asset.Value` that is also identified as important in Random Forest model.



(a) Gradient Boosting

(b) AdaBoost

Figure 15: Top 10 relative influence of variables from Boosting methods.

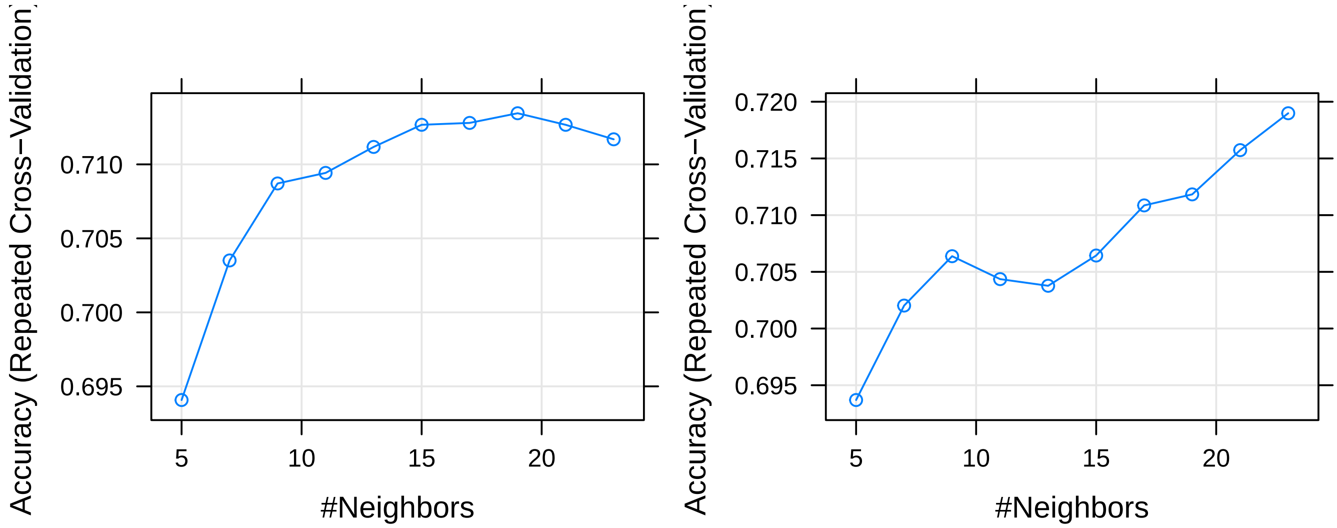
Neural Network

Neural Network techniques are used to recognise hidden patterns in the data and classify them. Another objective of applying Neural Network is to achieve dimension reduction by reducing the number of units in the hidden layer. Prior to fitting the models, the labels are transformed to be one-hot encoded. A 2-hidden-layer Neural Network is first fitted with input layer of 123 neurons and the middle layer of 60 neurons, with `epochs` equal to 500. The non-linear activation functions `sigmoid` and `softmax` are applied. The training history plot suggests that a smaller number of `epochs` can be used as validation accuracy begins to flatten at an earlier epoch.

This network is then compared with the other three networks: a network with 1 hidden layer of 60 neurons, a network based on the first model with 2 hidden layers but with smaller `epochs` equal to 100, and a network with 1 hidden layer of 30 neurons. By computing test errors, it shows that although the first model achieves rather high training accuracy at a large epoch, it results in lower prediction accuracy on the test set compared to models stopped at a small epoch. In addition, adding 1 hidden layer does not improve the training and test accuracy. However, when reducing the number of input variables from 123 to 60, the model achieves lower test errors. Thus, the second network using a hidden layer with 60 neurons that achieves the smallest test error rate among the fitted networks is used to compare with other classification methods.

kNN

The fact that there are a large number of highly correlated variables in the data may suggest that kNN is not appropriate to achieve accurate classification results. Nevertheless, we apply kNN to compare the results with other classifiers and the kNN model fitted on LDA. It is executed by tuning kNN models with 10-fold cross-validation and the tuning length of k is 10.
LDA is then applied before fitting a kNN model, in which case the dimension is reduced to 1 after projections. Figure 16 shows the accuracy against different k values. The model that is conducted LDA prior to kNN slightly increases the classification accuracy in the training process with an optimal k of 23 and accuracy of 0.719.



(a) kNN

(b) LDA+kNN

Figure 16: kNN model tuning process (CV accuracy against k nearest neighbors).

Compare the model performance on the test set

The six classification models based on Random Forest, Boosting, Neural Network and kNN are fitted on the test set to compare their performances assessed by test error rates and ROC curves. Table 1 shows the comparison of test errors and AUC values for various models. The ROC curves are plotted in Figure 17. The key findings are indicated as follows:

- Roughly speaking, there is no significant difference between all the models in terms of test error rates and AUC values, which is also verified by ROC curves.
- Random Forest and Boosting methods achieve an overall better performance on the test set considering both error rates and AUC values. In particular, Random Forest achieves the best with an error rate of 0.258 and AUC of 0.770.
- AdaBoost seems to have slightly higher error rate, compared to Gradient Boosting using Bernoulli distribution, which is also recommended in binary classification tasks.
- While Neural Network achieves dimension reduction, it performs the worst on the test set, with a test error rate of 0.286 and AUC of 0.714. However, it should be noticed that the parameters in Neural Network can be optimised through a process called hyperparameter.
- In terms of kNN method, as mentioned previously, applying LDA before using kNN slightly increases the classification performance, making it a more appropriate choice over solely using kNN.
- While Random Forest produces the best performance on the test set, it has a high computational cost, as well as that of Neural Network.

Table 1: Test error rates and AUC values for various models.

	Random Forest	Boosting (Binomial)	AdaBoost	Neural (60 units)	kNN	LDA+kNN
Error	0.258	0.267	0.269	0.286	0.283	0.284
AUC	0.770	0.766	0.768	0.714	0.744	0.748

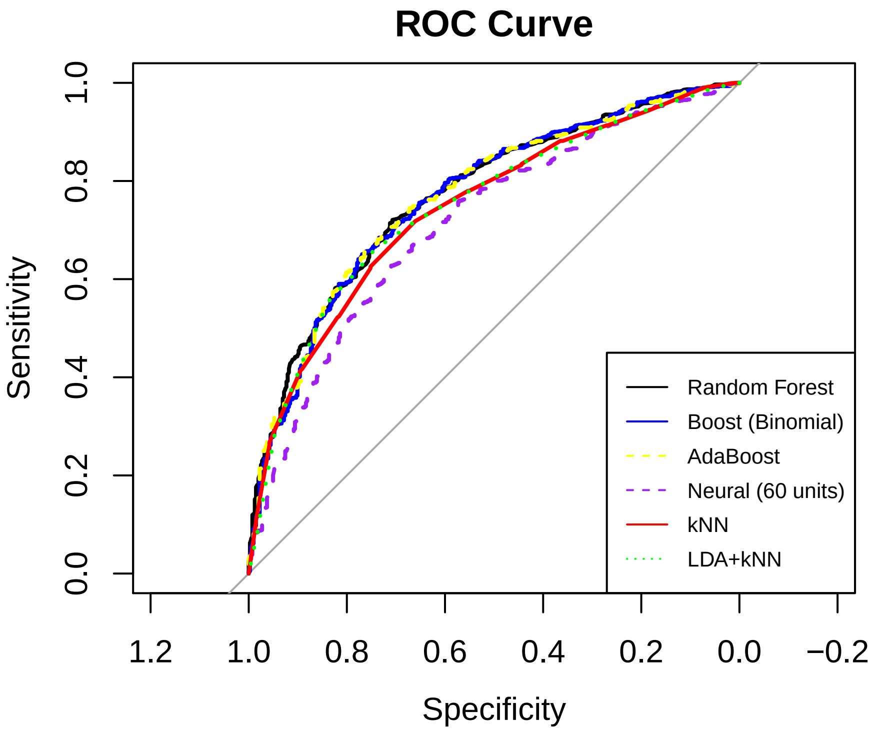


Figure 17: ROC curves from the test set for difference models.

The above analysis is conducted on all observations across 11 sectors. Considering that the influence of financial indicators on stocks may vary depending on the sector, the performance of classifiers may be influenced by the data itself. Therefore, in the subsequent analysis, a specific sector financial services is selected and the above classifiers are implemented on this subset.

2.3 Sectoral Analysis on Financial Services

In reality, it is difficult to predict stock price variation values based on a large number of financial indicators across different sectors. Therefore, the financial services sector is chosen as an example. As indicated in Figure 10, the financial services sector is the most imbalanced with 707 stocks in `Class 1` and 117 stocks in `Class 0`, the imbalanced ratio of which is 6.04. Based on previous analysis, the following classification models are chosen for comparison:

- Random Forest model tuned on a grid of `mtry` parameter using cross-validation. The optimal `mtry` is 2 with highest accuracy of 0.858;
- A Gradient Boosting model with the optimal number of boosting iterations equal to 123;
- A Neural Network with 1 hidden layer of 60 neurons for classification, of which the training accuracy is 0.873;
- Application of the upsampling method SMOTE to fit the kNN model. The optimal k value is chosen as 13 with the highest ROC of 0.760.

The comparison of the test error rates, AUC and ROC curves are indicated in Table 2 and Figure 18:

- As the models are fitted on a particular sector, i.e. financial services, there is greater difference in the overall performance of the classifiers, compared to the previous analysis.
- Random Forest and Gradient Boosting methods can still produce the best performance assessed by both test error rates and AUC values. Gradient Boosting is able to achieve the lowest test error rate of 0.134 and the highest AUC value at 0.828.
- It seems that the classification performance of Neural Network is improved in terms of test error with a value of 0.158. However, its AUC value is the lowest among various models at 0.694.
- It is worth noting that despite considering the upsampling method SMOTE, kNN method still achieves a relatively high error rate of 0.283. This is not surprising since kNN is generally not appropriate to conduct classification for high-dimensional data.

Table 2: Test error rates and AUC values for various models in financial services sector.

	Random Forest	Gradient Boosting	Neural (60 units)	kNN (SMOTE)
Test Error	0.138	0.134	0.158	0.283
AUC	0.823	0.828	0.694	0.759

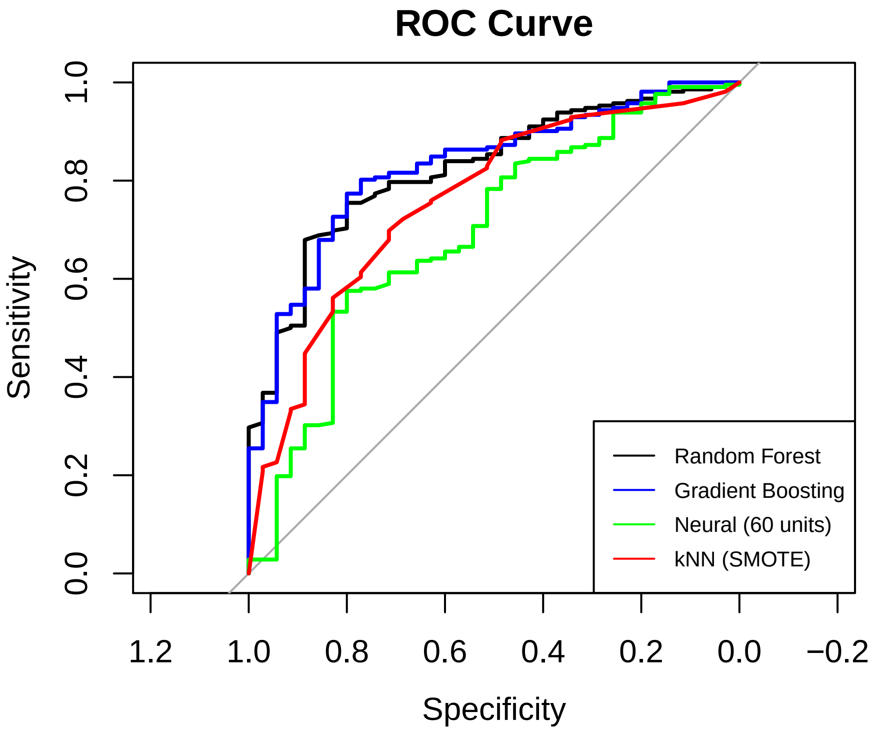


Figure 18: ROC curves from the test set for difference models in financial services sector.

2.4 Conclusion

Overall, Random Forest and Gradient Boosting perform the best on this high-dimensional data set, which contains a number of 123 financial indicators after pre-processing. Neural Network technique, though producing subpar results, might be optimised by selecting parameters via the hyperparameter process. However, the computational cost is quite high, similar to Random Forest. Generally speaking, kNN is not an ideal method for this classification task, as it does not perform well when the dimensions are high in the data, despite applying LDA first. Ultimately, when conducting classification tasks on stock data such as this, it is recommended to perform a per-sector analysis, as the nature of each sector is different, rather than conducting an analysis across all sectors. Essentially, the analysis should consider the different characteristics of the data depending on the objective to achieve for a classification task.