UTS

Assessment Task 2 – SAS Predictive Business Analytics

PREDICTIVE MODELLING

Group 1
Manjyot Joher | 12897981
Vi Nguyen | 13592629
Aiden Ye Yint Hlyan | 14017432
Su Myat Than Cin | 13486927
Ye Min Oo | 13506858

Executive Summary

With the increasing popularity of passive income strategies implemented to assist with long term secured savings, many banks are exploiting opportunities to launch direct marketing campaigns targeted at potential clientele. To align with this, the report examines records within the BANK_DIRECT_MARKETING dataset, which contains real-world examples of these campaigns, and is a subset of a larger dataset extracted from a Portuguese bank, over a time period of two years, from May 2008 to November 2010 (Moro et al., 2011). These records amalgamate data from 17 marketing campaigns targeted at convincing potential customers to enter a term deposit contract. By applying our data mining process to this dataset, we aim to improve the predictive capabilities of the Portuguese banking institution, allowing them to determine if a potential client is going to sign up for a term deposit, with a certainty of at least 75-80%. This is the main objective for the data mining process, and is crucial to support banks in forecasting the amount of incoming term deposits, so that corresponding lending calculations can be executed for other customers. Ultimately, we aim to assist the bank in streamlining future direct marketing campaigns, by determining which clients are more likely to subscribe to a term deposit.

This report proposes a systematic data mining approach to solve this problem, by identifying key business objectives, generating data exploration insights, justifying data preparation techniques, and executing comparative analyses of a plethora of model training iterations, to determine the best performing classifier. By conducting initial data exploration, we discovered key insights that explained skewed distributions, such as in the case of duration and balance attributes, which exhibited extreme kurtosis values of 7.36 and 119.65, respectively. We also took into consideration external factors at the time of data collection, to extract more enriching information from the dataset, by incorporating real world data relating to the global financial crisis, which occurred during the collection period of 2008-2010. To develop further insights to assist in guiding data preparation decisions, attribute distribution data was used to lay the basis for multivariate attribute comparisons to identify relevant relationships.

To directly address the binary classification task, thorough analyses of neural network, logistic regression, decision tree, random forest, gradient boosted tree, and support vector machine model iterations were conducted. Ultimately, we aim to achieve the highest performing model by rigorously testing each classifier via hyper parameterisation, whilst evaluating them against common performance metrics.

Table of Contents

Executive Summary *1*

1. Business Problem *4*

2. Report Structure *4*

3. Initial data exploration *4*

3.1. Data dictionary **4**

3.2. Attribute information, summary statistics & data distribution **5**

3.2.1. Education 5
3.2.2. Marital 6
3.2.3. Job 7
3.2.4. Age 7
3.2.5. Customer_ID 9
3.2.6. Default 9
3.2.7. Balance 9
3.2.8. Housing 10
3.2.9. Loan 11
3.2.10. Contact 11
3.2.11. Day 11
3.2.12. Month 12
3.2.13. Duration 13
3.2.14. Campaign 14
3.2.15. Pdays 15
3.2.16. Previous 16
3.2.17. Poutcome 17

4. Comparative Analysis *17*

4.1. Bivariate comparisons **17**

4.1.1. Age and Housing 17
4.1.2. Balance and Job 19
4.1.3. Balance and Duration 19
4.1.4. Campaign and Poutcome 20
4.1.5. Age and Education 21
4.1.6. Age and Balance 22

4.2. Multivariate comparisons **22**

4.2.1. Balance, Education and Job 22
4.2.2. Balance, Default and Education 23
4.2.3. Age, Balance, Job and Marital 23
4.2.4. Balance, Housing and Loan 24
4.2.5. Campaign, Contact, Duration and Poutcome 25
4.2.6. Balance, Pdays and Poutcome 25
4.2.7. Default, Poutcome and Previous 26
4.2.8. Age, Balance and Loan 27

5. Data pre-processing *28*

5.1. Data cleaning **28**

5.2. Data transformation **28**

5.2.1. Data type conversion 29
5.2.2. Binning 29

5.3. Dimensionality reduction **30**

5.4. General cluster analysis **31**

5.4.1. Correlation Matrix – SAS Visual Analytics 31
5.4.2. Relative Importance – Data Exploration Node 31
5.4.3. Variable Clustering Node 32
5.4.4. Feature Machine Node 33
5.4.5. Feature Extraction Node 34
5.4.6. Anomaly Detection Node 34

5.4.7. Clustering Node 34
6. Preprocessing Pipeline 35
7. Data mining methodology 35
7.1. Proposed data mining process 35
7.1.1. Stage One: Business Understanding & Problem Analysis 36
7.1.2. Stage Two: Data Exploration 37
7.1.3. Stage Three: Data Preparation 37
7.1.4. Stage Four: Model Training & Analysis 38
7.1.5. Stage Five: Comparative Model Evaluation 39
7.1.6. Stage Six: Model Deployment 40
8. Modelling techniques & experiment analysis 41
8.1. Neural Network 41
8.2. Logistic Regression 42
8.2.1. Forward Logistic Regression (FLR) 43
8.2.2. Backward Logistic Regression (BLR) 44
8.2.3. Stepwise Logistic Regression (SLR) 45
8.3. Decision Tree 47
8.4. Gradient Boosted Tree 48
8.5. Random Forest 49
8.6. Support Vector Machine 52
9. Comparative model assessment 53
10. Model deployment 58
11. Conclusion 59
12. Challenges encountered during exploratory data analysis 59
13. Appendix 61
13.1. Neural Network 61
13.1.1. Neural Network Configuration Table 61
13.2. Logistic Regression 61
13.2.1. Logistic regression parameter descriptions 61
13.2.2. FLR Configuration Tables 62
13.2.3. BLR Configuration Tables 64
13.2.4. SLR Configuration Tables 65
13.3. Decision Tree 66
13.3.1. Decision Tree parameter descriptions 66
13.3.2. DT Configuration Tables 66
13.4. Gradient Boosted 67
13.4.1. Gradient Boosted Tree parameter descriptions 67
13.4.2. GB Configuration Table 68
13.5. Random Forest 69
13.5.1. Random Forest parameter descriptions 69
13.5.2. RF Configuration Tables 69
13.6. Support Vector Machine 71
13.6.1. SVM Parameter descriptions 71
13.6.2. SVM Configuration Table 72
13.7. Assessment 2 – Work Distribution Table 72
14. References 74

1. Business Problem

With many banks wanting to accurately forecast their lending capabilities, it is vital that they are able to predict their long-term cash reserves to avoid borrowing above their threshold. To satisfy this requirement, many banks have exploited the opportunity to launch direct marketing campaigns targeted at potential clients, enticing them to signup for a fixed term deposit.

Unique records within the *'BANK_DIRECT_MARKETING'* dataset explicitly documents real-world examples of these direct marketing campaigns, and is a subset of a larger dataset extracted from a Portuguese bank, over a time period of two years, from May 2008 to November 2010 (Moro et al., 2011). The total number of unique records amounts to 10,578 entries. During this time, the bank documented data relating to 17 marketing campaigns, targeted at convincing potential customers to enter a term deposit contract, via in-house telemarketing (Moro et al., 2011). Using this dataset, our goal is to provide meaningful insights based on 18 input attributes such as job type, education, and marital status, mentioned in section 3.1 – data dictionary. Ultimately, upon completion of initial data exploration of attribute data, we aim to highlight key insights or trends within the data, and describe every attribute in respect to their distribution within the dataset. These steps will create the basis for further modelling carried out in the report, so that predictions can be made on which customers would be more likely to agree to a term deposit.

2. Report Structure

This report aims to present key insights and observations derived from a comprehensive examination of the selected *'BANK_DIRECT_MARKETING'* data set (UCI Machine Learning Repository, 2012), which will be documented below. The report is segregated into six major sections: data exploration, comparative analysis, data pre-processing, data mining methodology, modelling techniques and comparative model assessment. Within the data exploration section, attributes will be individually categorised, highlighting their distribution, along with key metrics. Building upon this, comparative analysis will delve deeper into the data by providing bivariate and multivariate relationship insights for certain attributes. These comparisons will provide a basis for any required pre-processing that will occur in the following section. The latter part of the report aims to outline the proposed data mining methodology for model training and comparison, so that the best performing model can be achieved.

3. Initial data exploration

In this section, attributes will be categorised into data types, and will have their summary statistics defined. Where relevant, summary statistics may be visualised to display interesting results.

3.1. Data dictionary

In preparation for further data analysis, attributes within the provided dataset will be classified into four types. These being:

Qualitative:

- **Nominal (categoric) –** attributes that have no significant meaning to them, other than being referred to as labels. Therefore, they cannot be ordered (e.g. Colours – 'Red', 'Blue', 'Green')
- **Ordinal –** attribute labels that can be ordered to show hierarchical meaning; however numerical operations such as addition and subtraction are not rational (e.g. Height – 'Tall', 'Average', 'Short')

Quantitative:

- **Interval –** attributes that can be ordered and are measured in fixed units. Subtraction is rational as the distance can be derived from two different values. Absolute zero values do not exist (e.g. Temperature in °F – '100°F', '50°F', '32°F')
- **Ratio –** attributes that can be ordered and are treated as real numbers, with an absolute zero value. All numerical operations can be conducted on these attributes. (e.g. Yearly income – '$155678.45', '$30000.00', '$22445.34')

The data dictionary below represents data types of input attributes, with possible modifications made, to better adhere to the business problem. Any changes to an attribute's type will be outlined in the pre-processing section of the report, following individual attribute exploration.

Attribute (in order of occurrence)	Data Type	Description	Example Value
Education	Nominal	Highest education level of client	Tertiary
Marital	Nominal	Marital status	Divorced
Job	Nominal	Occupation type	Student
Age	Interval	Age of potential client	25
Customer_id	Nominal	Unique row identifier	5122
Default	Nominal	Has the client ever defaulted on credit debt?	Yes
Balance	Ratio	Average yearly account balance (in Euros)	2500
Housing	Nominal	Does the client have an active housing loan?	No
Loan	Nominal	Does the client have an active personal loan?	Yes
Contact	Nominal	Communication type	Telephone
Day	Interval	Last contact day in the month (of the current/most recent campaign)	31
Month	Ordinal	Last contact month in the year (of the current/most recent campaign)	Jan
Duration	Interval	Duration of the last call with the client in seconds	245
Campaign	Interval	Number of contacts/calls conducted during this campaign (inclusive of the last contact)	6
Pdays	Ordinal	Number of days passed since last contact with client from a previous campaign	-1: target client was not previously contacted 1: target client was contacted in the last 1-273 days 2: target client was contacted more than 273 days ago
Previous	Interval	Number of contacts/calls for this client in the previous campaign	5
Poutcome	Nominal	Outcome of previous campaign for the client	Success
Target Variable: Y	Nominal	Has the client subscribed to a term deposit?	Yes

Table 1: Data Dictionary

3.2. Attribute information, summary statistics & data distribution

3.2.1. Education

Education is classified as a nominal attribute as it does not have an order, and is solely symbolic of the customer's highest education level. While it is possible to rank levels within this attribute ordinally from highest to lowest education, the order should not impact predictions for this business problem, and is therefore classed as nominal.

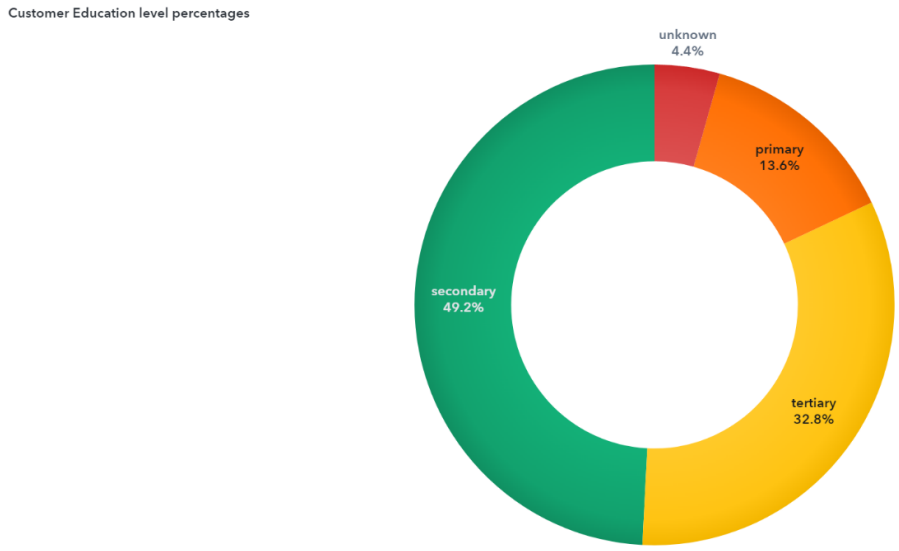


Figure 1: Education – Pie Chart

Prior to any pre-processing, it is interesting to note that the majority of clients in the bank's dataset have a secondary education as their highest academic achievement, amounting to almost 50%. It should also be noted that 4.4% of education values are marked as unknown, which could be indicative of a lack of reporting during the campaign. These are shown in red in Figure 1. To lower the number of dimensions in this attribute, the unknown category *may* be subsumed into the mode (most occurring) level for this segment, being secondary education.

3.2.2. Marital

Marital is classed as a nominal attribute, as ordinally ranking single, divorced, or married categories, does not provide additional value to the raw data. This attribute is representative of the marital status of each potential client, as reported to the Portuguese banking institution, during their telemarketing campaigns that ran from 2008-2010.

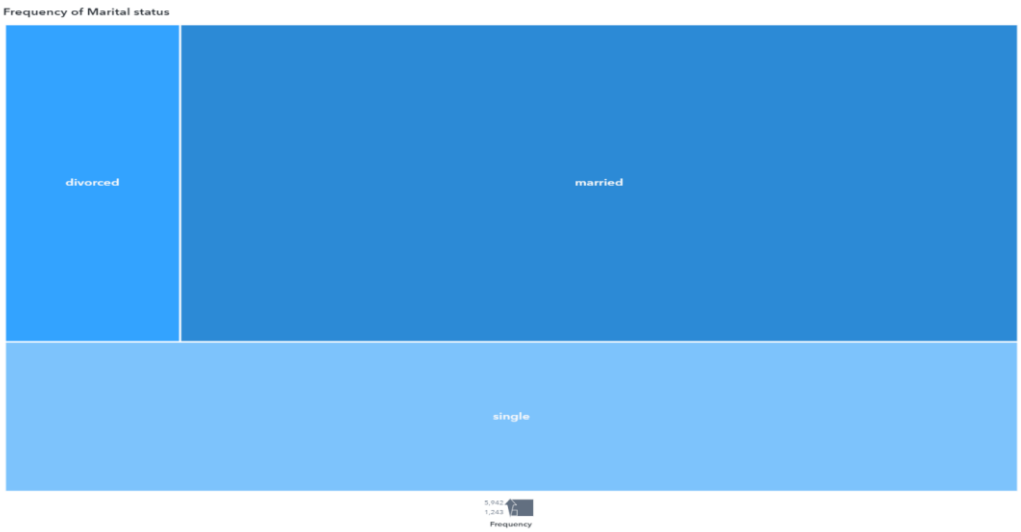


Figure 2: Marital – Pie Chart

Figure 2 above, visualises these 3 categories into a tree map. As shown, 'married' has the largest count amongst all potential customers, totalling to 5942. Contrastingly, potential clients who reported a 'divorced' marital status, had the lowest frequency, equating to 1243.This 378% increase from divorced to married, could be strategic within the bank's telemarketing campaign. It could be assumed that married customers have more disposable income to invest in a term deposit, in comparison to divorced individuals, and therefore were targeted more often. The latter category may have had to deal with expensive legal fees, wealth sharing between both parties, and may not be in the right mindset to open a term deposit. This notion will be further explored in section 4, comparing relevant attributes together such as, balance, age, and housing. Potential external factors such as the global financial crisis at the time of these campaigns, may have also influenced the subset of targeted clients sought out by the Portuguese banking institute.

3.2.3. Job

Job is a nominal attribute as it designates recorded occupation types during the bank's telemarketing campaigns. The tree map below illustrates occupation type, using frequency as its size, and gradient factor.

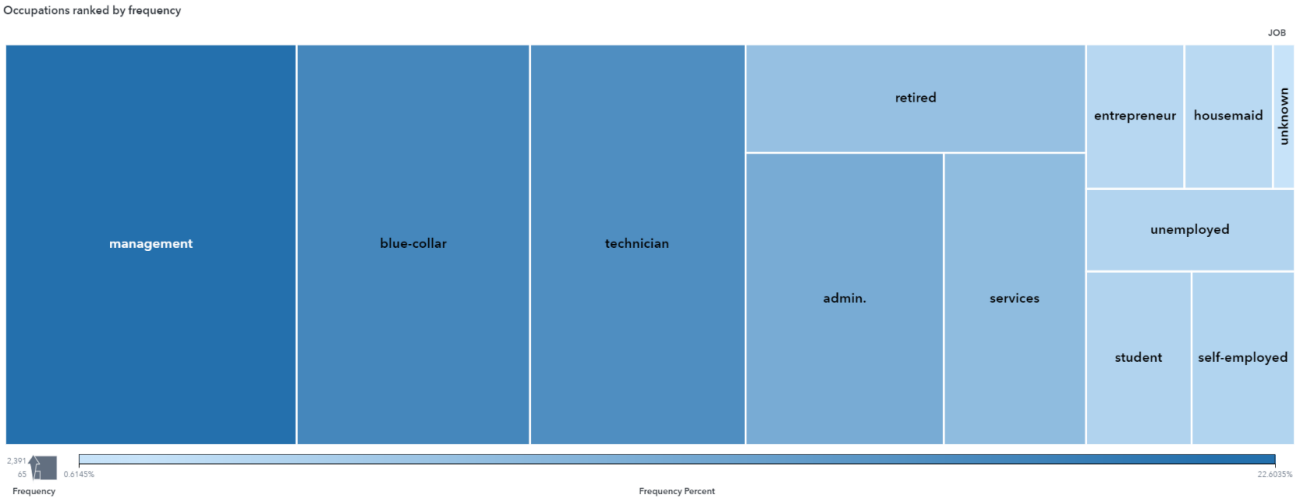


Figure 3: Job – Tree Map

JOB	Frequency	Frequency Percent ▾
management	2,391	22.60%
blue-collar	1,914	18.09%
technician	1,768	16.71%
admin.	1,185	11.20%
services	850	8.04%
retired	757	7.16%
student	375	3.55%
self-employed	367	3.47%
unemployed	353	3.34%
entrepreneur	291	2.75%
housemaid	262	2.48%
unknown	65	0.61%

Figure 4: Job – Tabulated Job Frequencies

The Job attribute contains 12 categoric levels, ranging from the largest being management at 22.6%, to housemaid being 2.48%. As depicted in the tree map in Figure 3, emphasis was placed on contacting potential clients with secure occupations such as managers, technicians, and administrators. Figure 4 reinforces this, with the majority of occupations below double-digit percentages (10% or lower), being mainly sole trader-based occupations. These may be assumed to be less financially secure, as retirees, students, housemaids and self-employed categories, may not have a steady stream of income. An exception to this is the 'service' category, which may have not been of great importance to the bank at the time. Given the global financial crisis that took place during these campaigns; service worker income may have become unstable. It can be assumed that the demand for service-based businesses dwindled during this period due to many people losing their own disposable income streams. Aside from this exception, the 'unknown' job category is representative of only 0.61% of the dataset, and could be due to incorrect recording during the collection process.

3.2.4. Age

Age is classified as an interval attribute, as it can be deduced that the difference between the ages of 20 and 30, is 10 years, which is the same difference between 30 and 40. While age can also be classified as a ratio, in this dataset there are no absolute 0 age values, as the minimum is 18.

Statistic:	Mean	Median	Range	Mode	Standard Deviation	Relative Variance	Skewness	Kurtosis
Value:	41.2268	39	95 (max) – 18 (min) = 77	31	11.9977	0.2910	0.8442	0.5614

Table 2: Age - Summary statistics

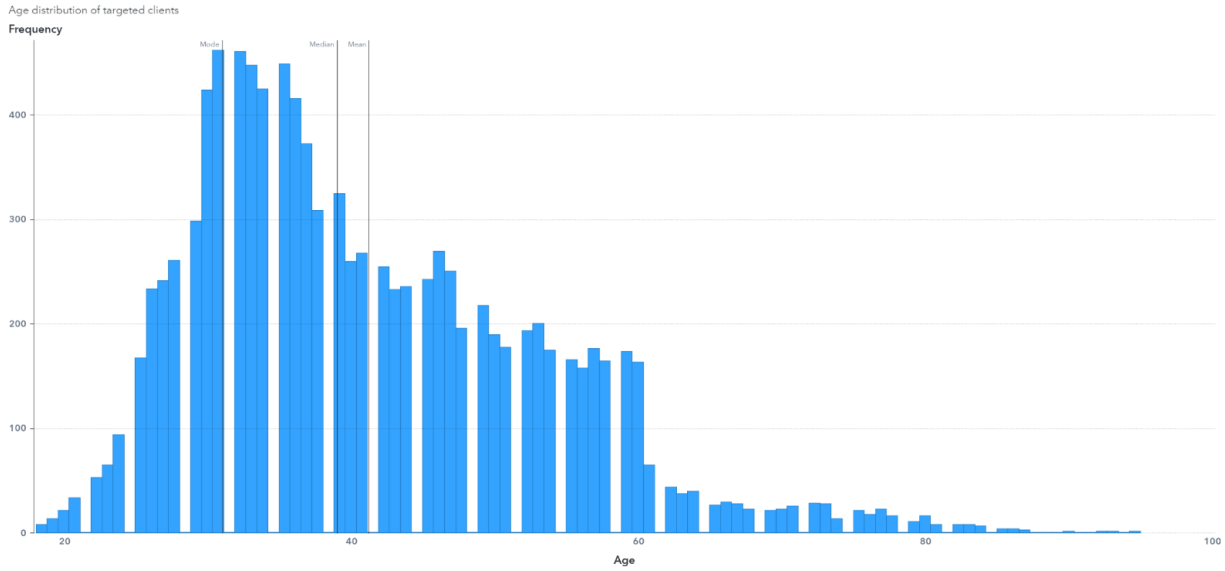


Figure 5: Age – Binned Histogram

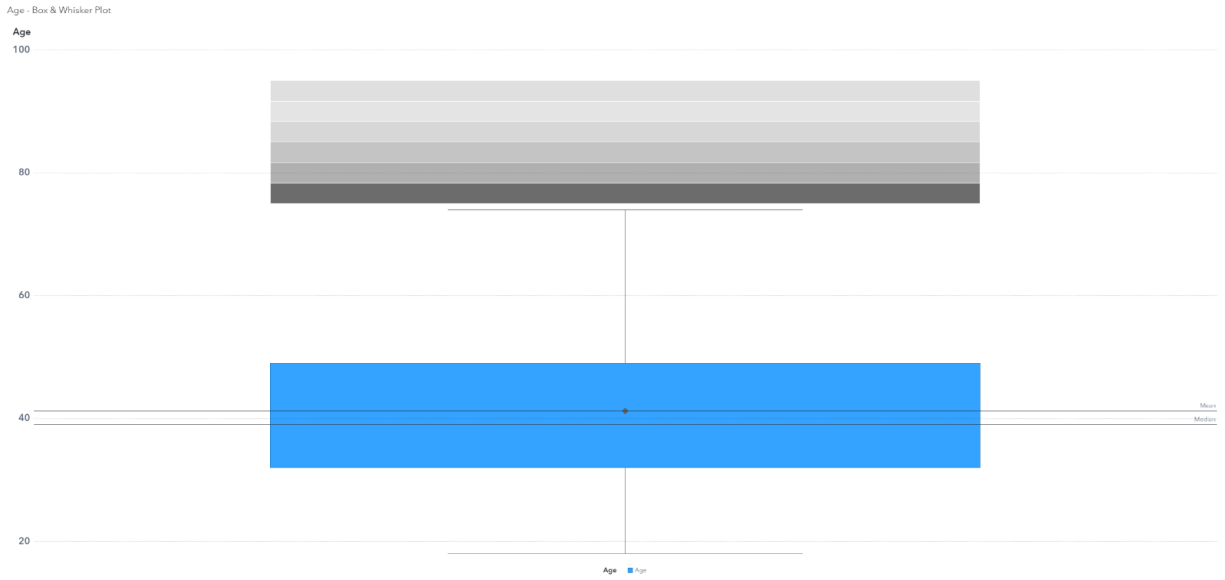


Figure 6: Age – Box & Whisker Plot

This attribute is representative of the age of potential clients targeted by the telemarketing campaigns, ran by the Portuguese banking institution. As shown in Table 2, the most occurring age amongst potential clients is 31, indicating that the distribution of ages is slightly positively skewed, with a skewness value of 0.8. This coincides with the big drop off in ages from 60 and older as shown in Figure 5, aligning with the positively skewed characteristic, of trailing outliers towards the right of the distribution (Klima, 2021, para. 2). Additionally, the kurtosis of 0.56, is indicative of relatively heavier tails, reinforcing the outliers shown visually in the positively skewed histogram in Figure 5 (Kenton, 2023; SAS, 2015). The box and whisker plot shown in Figure 6 also corroborates this narrative, and depicts a multitude of outliers shown in a grayscale gradient, above the third quartile of the chart. The least occurring outlier range shown is in the 92-95 bin, with a frequency of 6. The most occurring outlier range is the 75-80 bin, with a frequency of 80.

Figure 7: Age – Missing Values

It is important to note that Age is the only attribute in the entire dataset that has missing values. Using the data exploration node in SAS Model Studio, it is evident that this equates to 24 missing values, documented by the percentage in Figure 7. These values will be addressed in section 5.1 – data cleaning. It should also be noted that both Figures 5 & 6 have been illustrated with binned age values, with the histogram being capped at 100 bins due to SAS Viya limitations.

3.2.5. Customer_ID

'Customer_id' is classified as a nominal attribute as it does not have any meaning aside from labelling each row in the dataset. The number of rows within the dataset equates to 10,578. This attribute will be omitted from training, validation, and test sets during pipeline creation. Its removal will be addressed further in section 5.3 – dimensionality reduction.

3.2.6. Default

Default is classified as a nominal attribute which indicates whether a client has defaulted on credit debt.

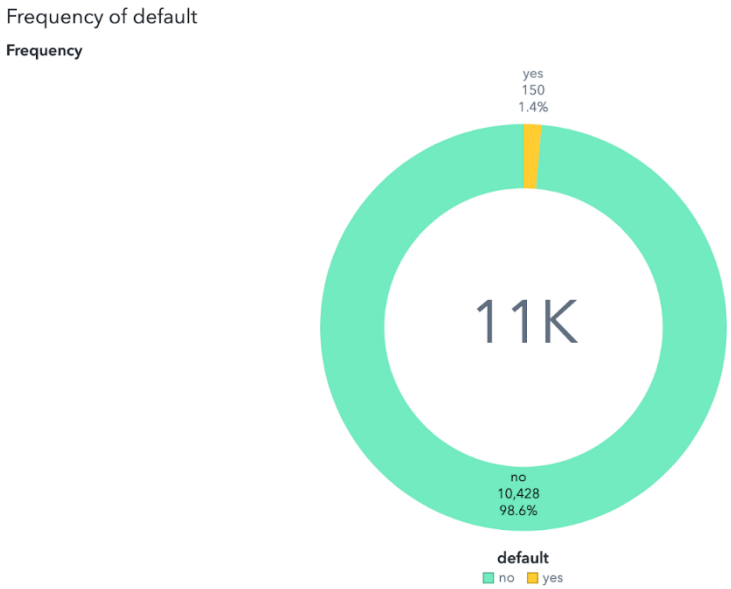


Figure 8: Default – Pie Chart

According to Figure 8, 10,428 people (98.6%), have not missed payments on credit debt, contrasted with 150 people that have. The incredibly low default rate of 1.4% is significant, showing that the vast majority of Portuguese clients in this sample are financially responsible, and are effective at managing their credit debt. This is in the best interest of the Portuguese institution, so it is logical that this distribution of target clients was chosen for the marketing campaign.

3.2.7. Balance

Balance is a ratio attribute which indicates the potential client's average yearly balance in euros. It is identified as a ratio attribute as it has a logical zero value in the dataset. Moreover, the attribute has multiplicative and divisive properties that enable balances such as 600, to represent a value twice the size of 300 - this can be useful in explaining differences between data points.

Statistic:	Mean	Median	Range	Mode	Standard Deviation	Relative Variance	Skewness	Kurtosis
Value:	1,548.5298	566	81,204 (max), -3058 (min)	Not Applicable (unique value per row)	3,130.5653	2.0216	7.7168	119.6499

Table 3: Balance – Summary statistics

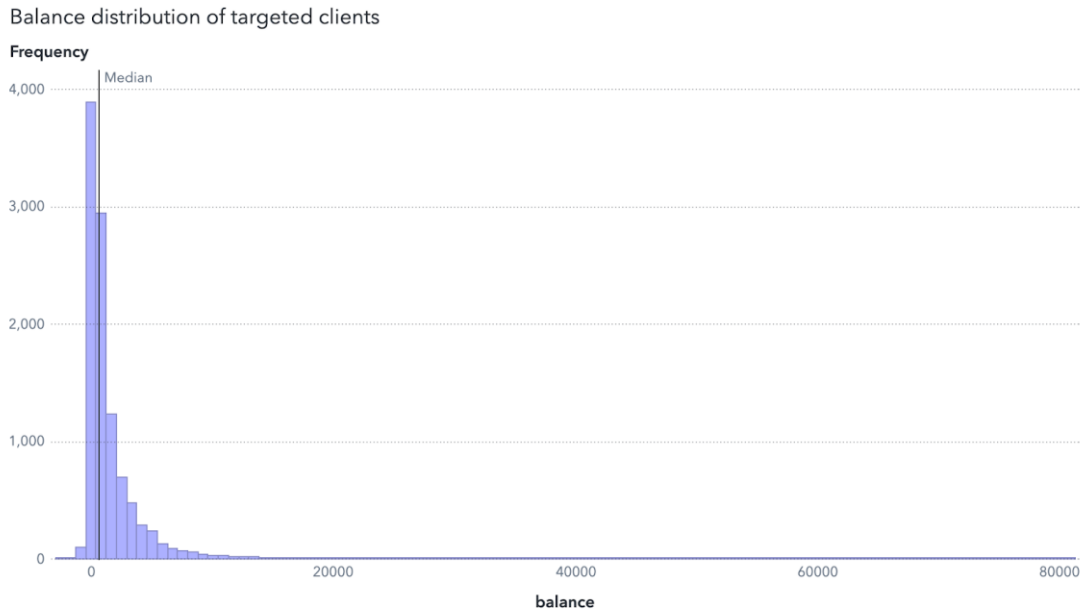


Figure 9: Balance – Binned Histogram

As taken from Table 3, the kurtosis value of 119.65 is significantly higher than a normal distribution, characterising the distribution as platykurtic (Frost, 2022). In contrast to a normal distribution, platykurtic distributions contain heavy tails and a sharp peak, which show that there are more outliers present (Glen, n.d., para. 1). This characteristic is clearly evident in the distribution shown in Figure 9. When referring to bank balances, this indicates that a disproportionately large percentage of customers have very high average yearly balances. This is corroborated by the significant positive skew of the distribution, equating to 7.72. Positive skewness means the tail is longer on the right side than the left. This indicates that the majority of targeted clients have balances that are lower than the mean, with a small number of them having extremely high balances that are pushing the mean upward. For relative variance, a value greater than 1 (in this case, 2.02) denotes a high degree of dispersion within the data.

3.2.8. Housing

Housing is a nominal attribute and is indicative of the client's housing loan status, and is therefore a binary variable.

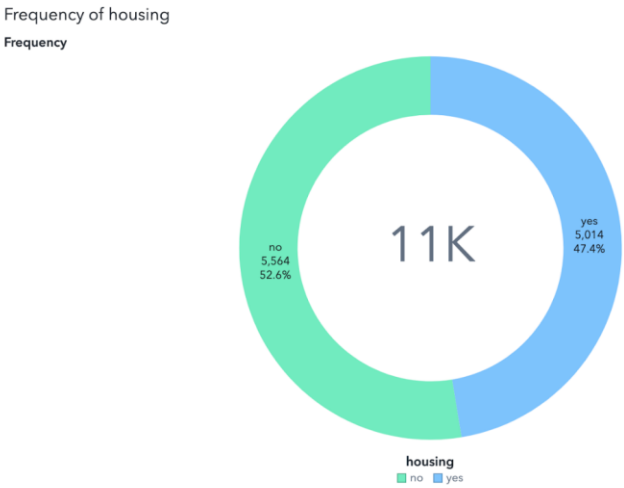


Figure 10: Housing – Pie Chart

As shown in Figure 10, the results indicate that 5,564 people (52.6%), do not have an active housing loan and 5,014 people (47.4%), hold an active one. The data shows a nearly equal distribution of clients with and without active mortgages. During comparative analysis, it would be interesting to look at how old these two groups are. For example, while older age groups may have paid off their mortgage loans, younger people may have not.

3.2.9. Loan

Loan is characterised as a nominal type, capturing whether a client has an active personal loan or not. Upon analysing the dataset, an overwhelming majority of clients, precisely 87.1% or 9,210 individuals, have reported not having an active personal loan, as visualised in Figure 11. Contrarily, 1,368 clients, representing 12.9% of the dataset, confirmed the presence of a personal loan. This significant skew towards the "no" category indicates that the majority of potential clients approached by the bank's telemarketing campaigns do not have other personal loan commitments, potentially making them more financially flexible.

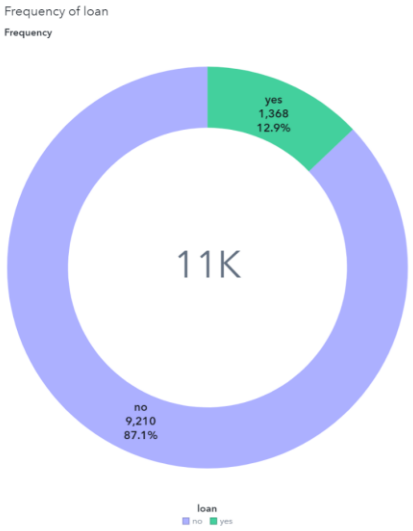


Figure 11: Loan – Pie Chart

3.2.10. Contact

'Contact' is delineated as a nominal type, indicating the mode of communication through which clients communicate with the bank. This attribute is comprised of three distinct categories: cellular, unknown, and telephone. Within the dataset, a dominant majority of 7,682 clients, equivalent to 72.6%, have opted for cellular communication. In contrast, a notably smaller fraction, 712 clients or approximately 6.7%, have chosen the telephone as their mode of contact. Interestingly, there's a sizable segment, about 20.6% or 2,184 clients, where the mode of communication remains 'unknown'. This category can be attributed to several factors such as data entry errors, or perhaps the possibility of utilising alternate contact methods, not specified in the template used during initial data collection. To potentially improve model predictions, considering strategies to handle this 'unknown' category, such as imputation or grouping may be employed, if it's deemed important in swaying the outcome of the dependent variable, '*y*'.

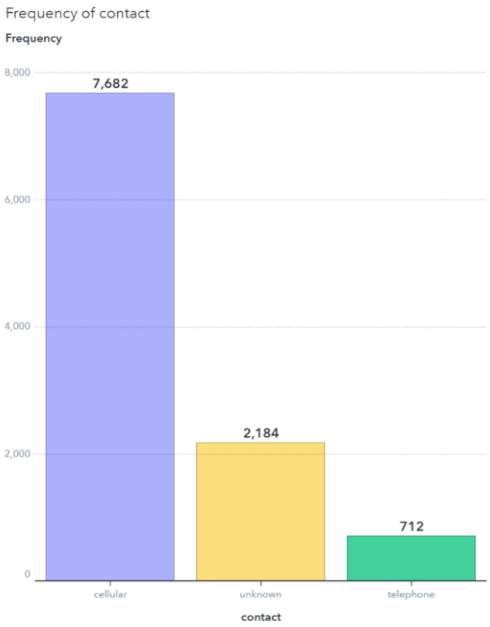


Figure 12: Contact – Histogram

3.2.11. Day

The 'Day' attribute is representative of the last contact day in the month of the most recent campaign. Day is an ordinal data type, as it is meaningful to rank them categorically, in accordance with the Gregorian calendar. As shown in Figure 13, the call frequency ranges from 110 to over 500. On most days, the frequency is in the range of

300 to slightly over 400. There are a few days with abnormally low frequency, such as the 1st, 10th, 24th and the 31st, where the frequency drops below 200.

Statistic:	Mean	Median	Range	Mode	Standard Deviation	Skewness	Kurtosis
Value:	14.4759	15	31 (max), 1 (min)	20	8.4138	0.1294	-1.0608

Table 4: Day – Summary statistics

As shown in the summary statistics table above, a skewness value of 0.1294 indicates that the 'Day' attribute has a very negligible positive skew. This slight skew is more characteristic of a relatively symmetrical distribution, rather than a strong peak. This is also validated by the negative kurtosis value of approximately -1, stipulating relatively even tails, with no explicit outliers, as shown in Figure 13.

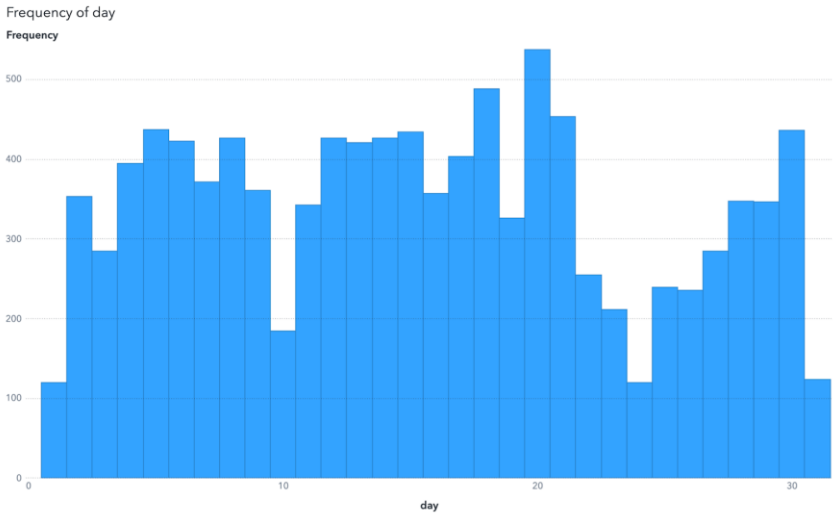


Figure 13: Day – Histogram

Moreover, the frequency of contacts drops relatively low at the start and end day of the month. This reduction could indicate that the customers refused to interact with the firm, or the bank was running on skeleton staff during these periods. Additionally, the contact frequency spikes on day 20, with approximately 537 contacts, followed by day 18 with 488 contacts. The increase in contacts on these days may be a result of potential clients receiving their monthly wage, based on industry trends; and therefore have more resources allocated towards contacting them. These spikes are followed by an off-peak period from the 22nd to the 26th, symbolising the only period of relatively low-frequency contact.

3.2.12. Month

The 'Month' attribute is indicative of the last contact period within the year. This attribute is classified as ordinal, as it follows the standard Gregorian calendar.

Statistic	**Value**
Mode	May (Frequency: 2,603)

Table 5: Month – Summary statistics

Timing is crucial when it comes to campaign deployment and data reconciliation. The selected contact period may significantly enhance the engagement of the potential consumer; and attract a greater response rate, thereby increasing the probability of signing up for a term deposit. A few factors that the Portuguese banking institution may have considered when planning telemarketing campaigns include: the seasonal trends in local markets, consistency of campaigns from 2008-2010 and the global financial crisis.
May accounted for the most contacts, with a value of 24.6%, as shown in Figure 14 below. Subsequent months, June, July, and August, also have high contact frequencies in the year. These four months contribute to nearly two-thirds of the bank's total calls in a year. In contrast, there were a few months when the communication between both

parties reduced. These months were September, October, December, January, and March, and only contributed to approximately 15% of the total call frequency.

Considering the ordinality of the data, and the fact that these real-world campaigns were conducted during the global financial crisis, it is important to also consider external factors that may have influenced the monthly contact distribution. While the contact year is not provided within the dataset, it may be assumed that the high contact frequencies in May, could have been due to upward-trending deposit interest rates, as shown in Figure 15. This trend marks the beginning of the recovering Portuguese economy towards the start of 2010, in which the last marketing campaigns were conducted. It may be assumed that the majority of contacts made in May, June, July and August took place in 2010; where potential clients could be motivated by higher interest rates (TheGlobalEconomy.com, 2023), and therefore justifies the bank's motivation to increase call frequencies.

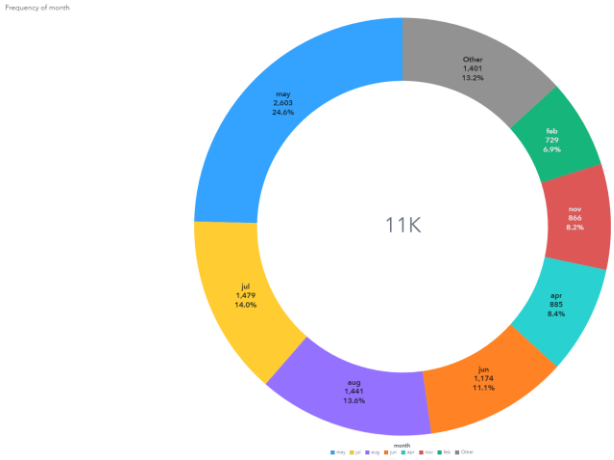


Figure 14: Month – Pie Chart

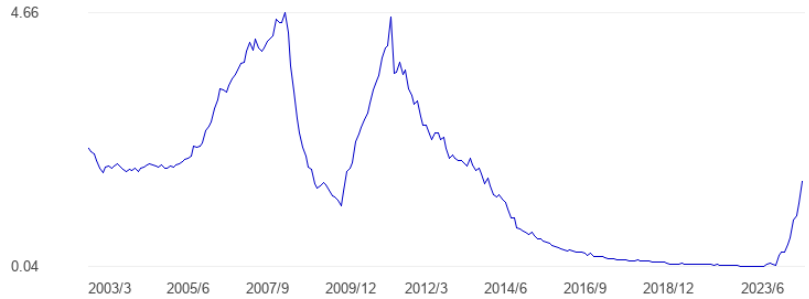


Figure 15: Historical deposit interest rates in Portugal – Line Graph (TheGlobalEconomy.com, 2023)

3.2.13. Duration

Duration is designated as an interval attribute, due to it representing the duration (in seconds) of the previous call with the potential client. This allows differences between two or more call durations to be made, aiding in comparative data point analysis.

Statistic:	Mean	Median	Range	Mode	Standard Deviation	Relative Variance	Skewness	Kurtosis
Value:	379.2437	260	3,881 (max) – 4 (min) = 3877	1,086 calls in bin: 120.31 – 159.08 seconds	350.9718	0.9255	2.1468	7.3608

Table 6: Duration – Summary statistics

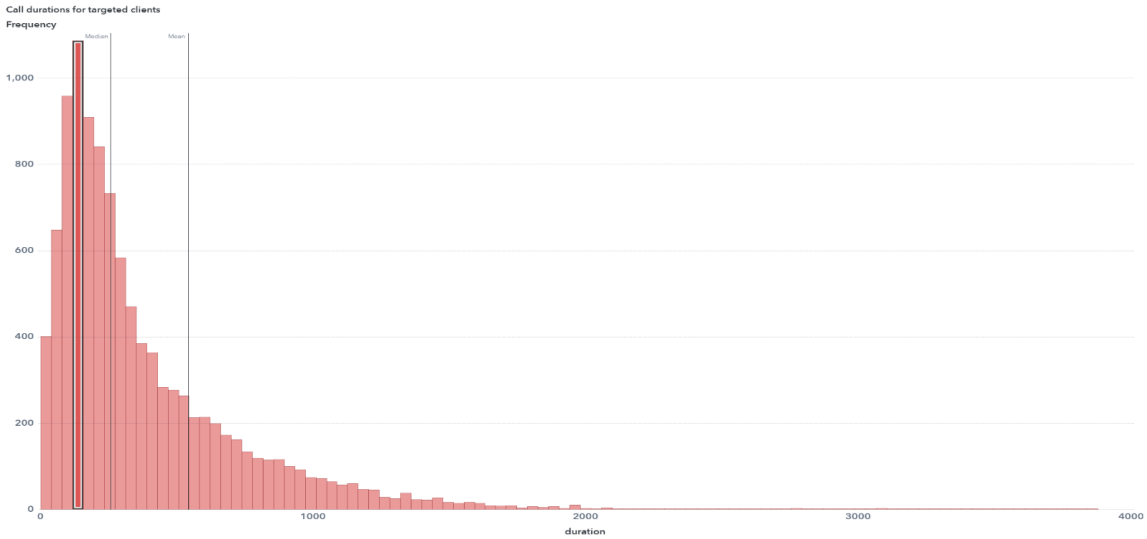


Figure 16: Call durations for targeted clients – Binned Histogram

Figure 17: Duration – Box & Whisker Plot

Call duration may be valuable in determining if targeted clients sign up for a term deposit, as greater call times may provide insight into client fatigue. Figure 16 depicts the distribution for contacted clients, with a noticeable peak towards the left of the chart. This is validated by the skewness value of approximately 2.15, indicating a strong positive skew. Contributing to this skew is the binned mode of approximately 120 – 159 seconds, representing a frequency of 1086. This highlights the fact that most calls to potential clients only last between 2 - 3 minutes, perhaps establishing a premise that most people do not need a long amount of time to decide on a term deposit. This notion will be further examined in section 4 – comparative analysis. Additionally, the histogram illustrates the decreasing graduation of bins towards the tail end of the graph. This consistently dropping frequency is also representative of the relatively high kurtosis value of approximately 7.36. This abnormally high concentration of outliers, when compared to a normal distribution, is also evident in Figure 17. The box plot depicts these outliers in a grayscale gradient above the third quartile, with the darkest section being indicative of the most occurring outlier bin (1052 – 1523.5 seconds), with a frequency of 450. It could be assumed that these duration outliers represent clients with lower average bank balances, and therefore need more time to decide on signing up for a term deposit. This supposition will also be considered in section 4.

3.2.14. Campaign

Campaign is an attribute representing the number of contacts performed during this campaign for the potential client. It is defined as an interval attribute, as there are no 0-values present, or these records would simply not exist in the dataset, as they were never contacted. This is confirmed by the minimum value of 1.

Statistic:	Mean	Median	Range	Mode	Standard Deviation	Relative Variance	Skewness	Kurtosis
Value:	2.4748	2	50 (max) – 1 (min) = 49	1	2.6152	6.8392	5.0976	44.6295

Table 7: Campaign – Summary statistics

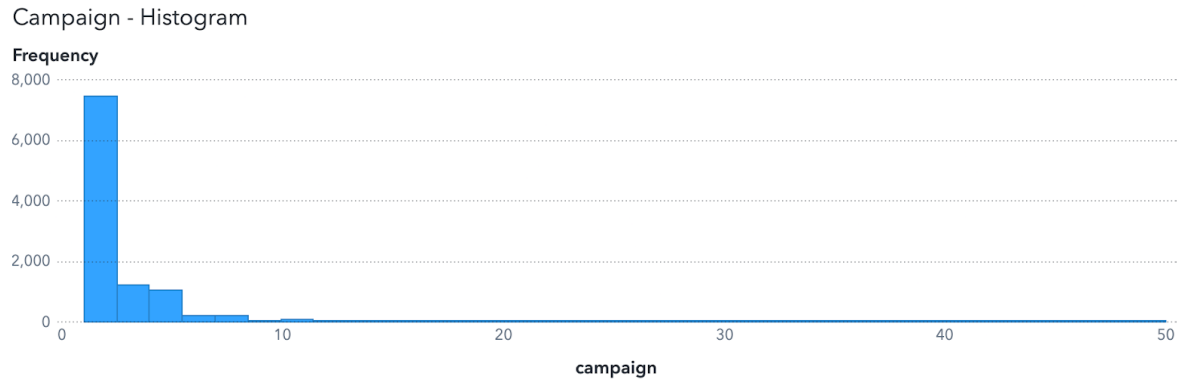


Figure 18: Campaign – Histogram

Table 7 shows the summary statistics regarding the attribute. The values range from 1 to 50, with the median being 2. Figure 18 depicts that the majority of contacts within this campaign are under 10, and is indicative of a positive skew.

3.2.15. Pdays

PDays refers to the number of days that have passed after a potential client was last contacted from a previous campaign. The Pdays attribute can be regarded as ordinal, as it includes distinct categorical values that can be ordered logically. These are tabulated below, and given their tiered hierarchy of durations, there is a clear logical order that can be followed.

PDays	Frequency	Frequency Percentage
-1 (Not Contacted)	7866	74.36%
1 (contacted within the last 1 - 273 Days)	1109	10.48%
2 (contacted more than 273 Days ago)	1603	15.15%

Table 8: Pdays – Summary statistics

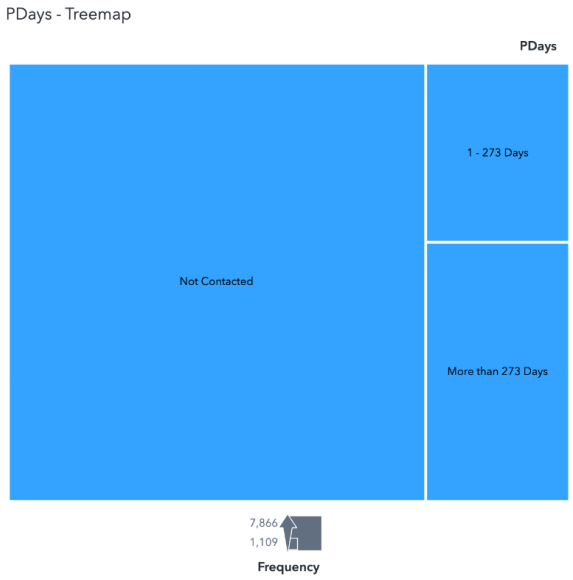


Figure 19: Pdays – Treemap

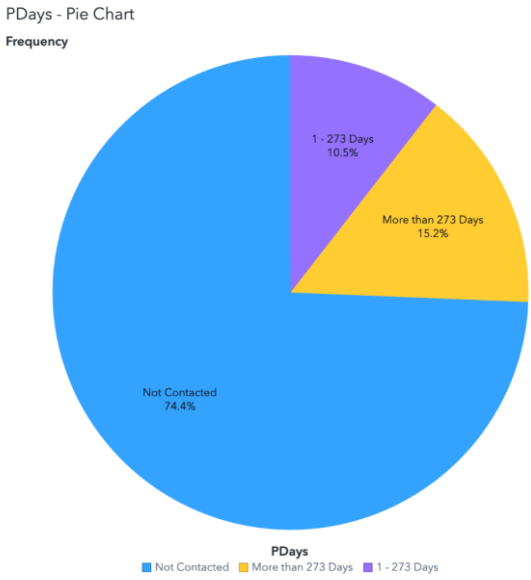


Figure 20: Pdays – Pie Chart

The tree-map presented in Figure 19 highlights the dominance of potential clients that were never contacted in a prior campaign to this one. The pie chart above also reinforces the idea, as this category accounts for 74.4% of the entire dataset. This provides insight into the bank's marketing strategy, suggesting that potentially untapped clientele, may be more willing to register for a term deposit, as they were never previously contacted before the current campaign.

It appears that those customers who were contacted a long time ago (more than 273 days), make up 15.2% of the set, which is bigger than the 10.5%, which is made up of customers who had more recent contacts. Customers who had been contacted previously may have biases towards the bank, which may influence their decision on whether to subscribe a term deposit.

3.2.16. Previous

This attribute refers to the number of contacts made before the current marketing campaign. It has values ranging from 0 to 275. It is considered as interval data, as a true zero point in this case provides no further statistical meaning, aside from representing that call frequency in a previous campaign was 0 for a particular target client.

Statistic:	Mean	Median	Range	Mode	Standard Deviation	Relative Variance	Skewness	Kurtosis
Value:	0.8525	0	275 (max), 0 (min)	0	3.4721	12.0556	48.5081	3694.2714

Table 9: Previous – Summary statistics

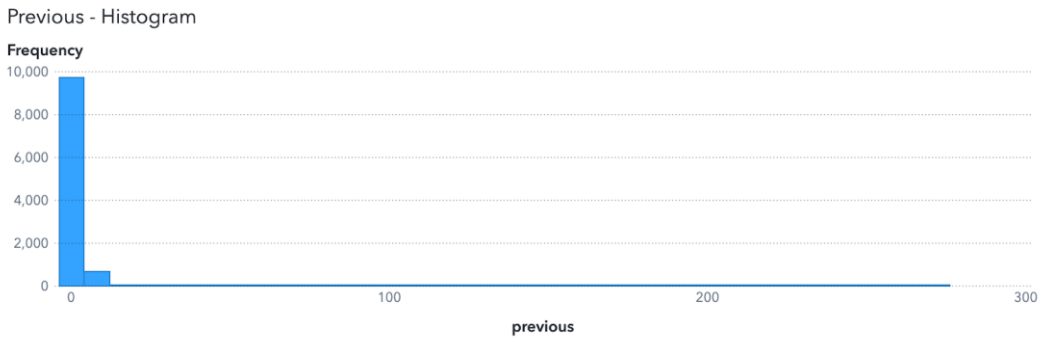


Figure 21: Previous – Binned histogram

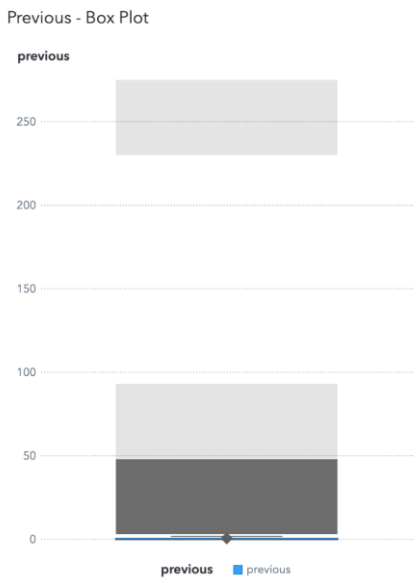


Figure 22: Previous – Box & Whisker Plot

In Table 9, the summary statistics show that the median and mode equate to 0, which proposes that a significant portion of target clients had no interactions before the current campaign. Figure 21 demonstrates the high positive skewness of data, with a value of 48.5, and an extremely high kurtosis of 3694, indicating the occurrence of

significant outliers. This is corroborated by the box plot, showing that most target clients were contacted between 1 and 50 times. It is also noteworthy to look at the one outlier who was contacted 275 times prior to this campaign, and resultantly, played a major role in the high kurtosis value.

3.2.17. Poutcome

Poutcome is an attribute that represents the outcome of the previous marketing campaign. It is categorical in nature and contains values tabulated below. Given the nature of these categories, they don't have a clear order which can be meaningful. While it could be argued that 'failure' is before 'success' in terms of measuring campaign success, 'other' and 'unknown' values don't have a clear position. Therefore, *poutcome* is a nominal attribute.

Poutcome	Frequency	Frequency Percentage
Success	1055	10%
Failure	1153	10.9%
Other	502	4.7%
Unknown	7868	74.4%

Table 10: Poutcome – Summary statistics

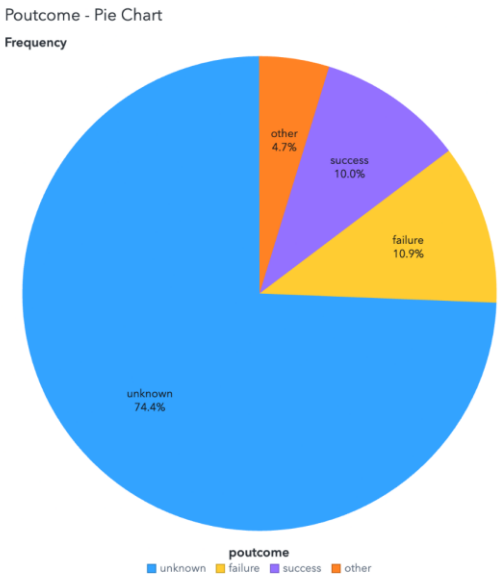


Figure 23: Poutcome – Pie Chart

It can be seen that 74.4% of the values are categorised as 'unknown', indicating that the majority of the outcomes from the previous marketing campaign are unknown. Both 'success' and 'failure' outcomes have similar frequencies, with 10.0% and 10.9% respectively. This suggests that when the outcome of the previous marketing campaign is known, there is a an approximate even split.

4. Comparative Analysis

Comparative analysis is a vital step within the exploratory data analysis process, as it can unravel key insights into relationships between chosen variables. This can improve dataset literacy, and assist in making informed pre-processing decisions, as outlined in section 5. By examining key relationships between attributes, potential trends and correlations can be derived, painting a narrative on the factors surrounding the telemarketing dataset, even considering external factors at the time of data collection.

4.1. Bivariate comparisons

4.1.1. Age and Housing

Age and Housing are assumed to be two important attributes when it comes to having free cashflow. This is especially important when attempting to persuade target clients to sign up for a term deposit, essentially locking their funds away for a specified period of time. Figure 24 below combines these two attributes into a single scatter plot to assist in correlation interpretation.

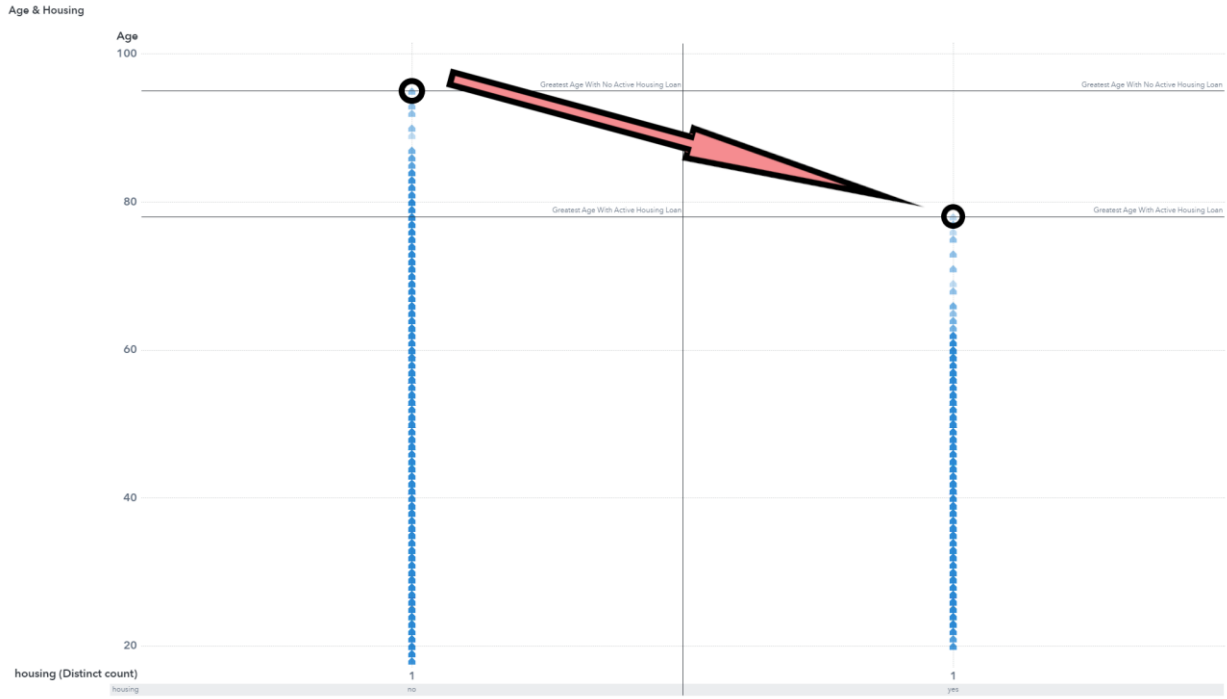


Figure 24: Age & Housing – Scatter Plot

As shown by the descending arrow, it is evident that there is a weak trend portraying that older clients have no active housing loans, whilst younger clients do. The blue colour gradient displayed on the house icons, indicate the concentration of ages amongst the overall distribution. As shown, there is a higher concentration of ages 20 – 60 for those that have an active housing loan, and 20 – 80 for those that do not. Two specific points have also been highlighted by black circles. The leftmost circle represents the greatest recorded client age who does not have an active housing loan, whilst the rightmost circle shows the greatest cage of a client who does. These points equate to a value of 95 and 78 respectively. This again corroborates the narrative that older targeted clients have paid off their mortgages and younger clients have yet to do so.

Additional external factors at the time of data collection may have also played a role in this relationship. The BANK_DIRECT_MARKETING dataset was assembled during the global financial crisis, by collating 17 telemarketing campaigns from May 2008 to November 2010. During this time, it was likely that free cashflow plummeted for residents and those with active home mortgages had to defer payments or overdraft credit accounts to continue making payments. This financial instability could've influenced this data for those with active loans, showcased by the 24 clients who were between the ages of 63 and 78 (inclusive of both). The pension age in Portugal in 2008 was 61.5 years, allowing clients above or at this age to receive their payments (Campos & Pereira, 2008, p. 44). This external data assists in classifying these 24 clients as outliers, as even with an assumed pension, they still have an active mortgage. Furthermore, it is highly unlikely that these clients within the 63-78 age range were purchasing new houses during the global financial crisis.

During this period, Portugal was also going through pension reform, by setting a transitionary period to slowly increase the retirement age, and therefore the pension payouts (Campos & Pereira, 2008). During the telemarketing campaign, the retirement age increased from 61.5 to 62.5, in half-year increments. This may have also impacted those that were close to retirement age, but still had an active mortgage, making them less enthusiastic to sign up for a term deposit. This is corroborated in Figure 25 below, where the people with active mortgages who agreed to a term deposit, are proportionally less than those who did not have one. This characteristic has been highlighted by a black oval encapsulating the column's data points.

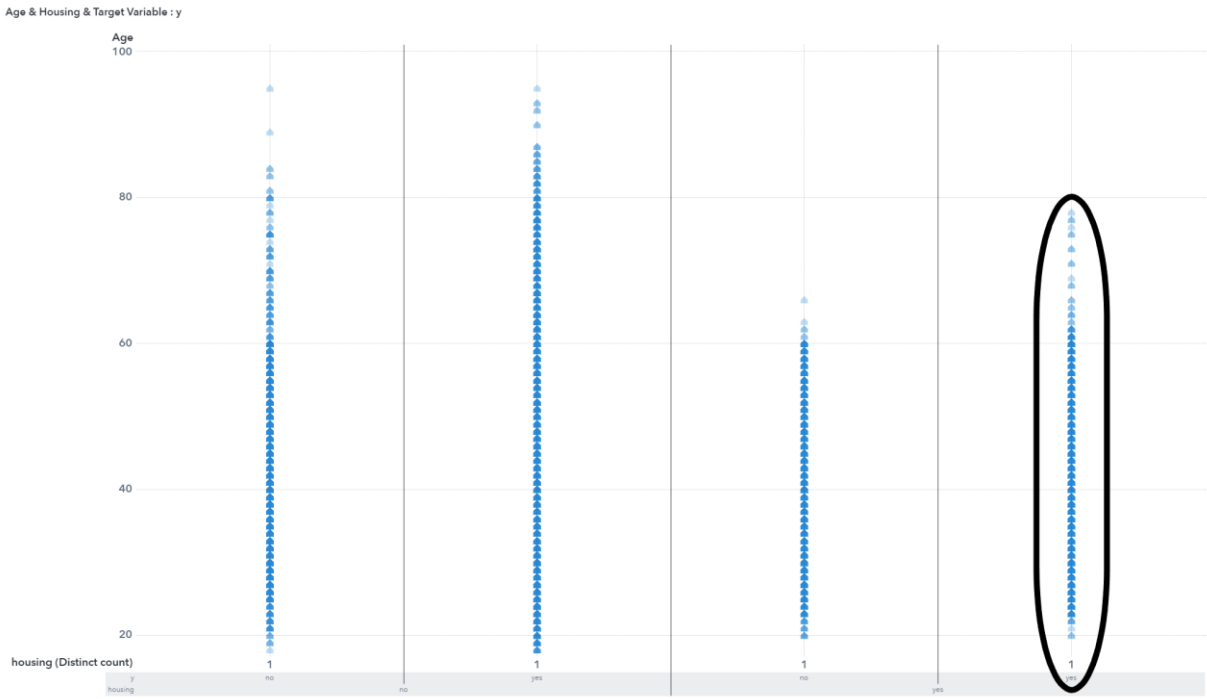


Figure 25: Age, Housing & Target Variable: Y – Scatter Plot

4.1.2. Balance and Job

The scatter plot depicts the association between the average annual account balance (in Euros), and the kind of occupation. The plot moderately suggests that those with higher-paying jobs such as managers and technicians tend to have higher average yearly account balances. Interestingly, retirees also tend to have relatively high average bank balances, which may represent their access to an influx of cash flow from their retirement pensions. Holistically, the average yearly account balance and occupation type do not have a generalised linear relationship. The average yearly account balance does not increase linearly with occupation type. For example, the average yearly account balance for management is not twice the average salary of a housemaid, and therefore the relationship is largely comparative, rather than direct.

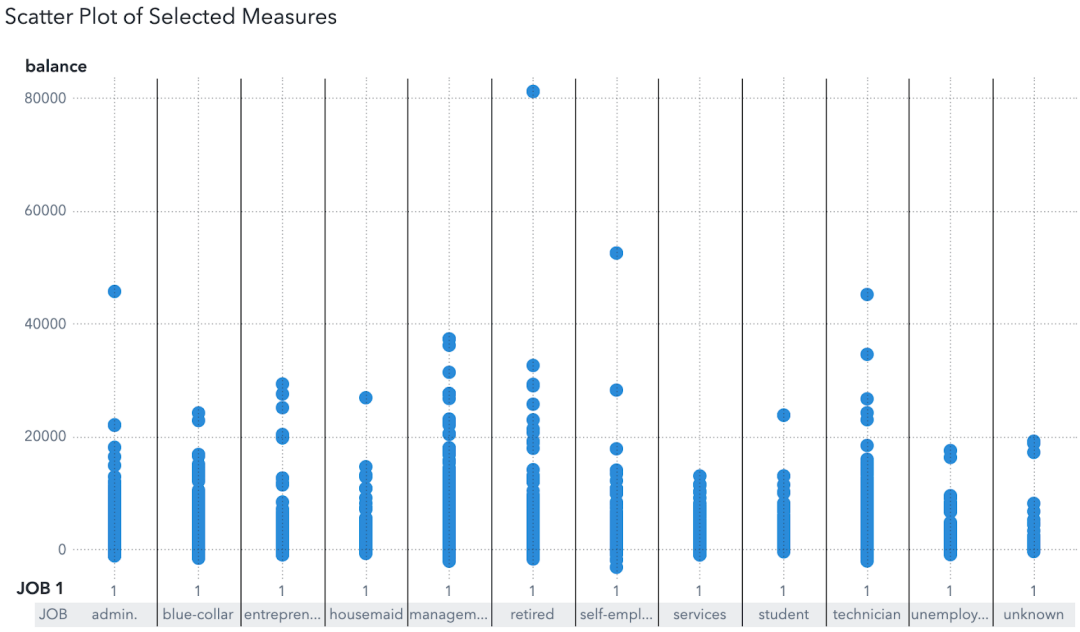


Figure 26: Balance & Job – Scatter Plot

4.1.3. Balance and Duration

The scatter plot provides insight into the relationship between a client's average yearly account balance (measured in Euros) and the duration of their last call with the bank (in seconds). The x-axis of the plot represents the duration, while the y-axis captures the balance. From the plot, it's evident that there's a positive correlation between the two

variables, suggesting that clients with higher balances generally spend less time on calls. This tendency might be due to their financial flexibility, enabling them to make quick decisions whether to sign up for a term deposit, as they may have a plethora of other investment strategies. Contrastingly, there are of several outliers that indicate an inverse relationship. Notably, there are points that represent target clients with low balances but with lengthy call durations, as depicted by the tail towards the bottom right of the plot. Such patterns may indicate clients facing financial difficulties and seeking assistance on how effective a term deposit may be for them.

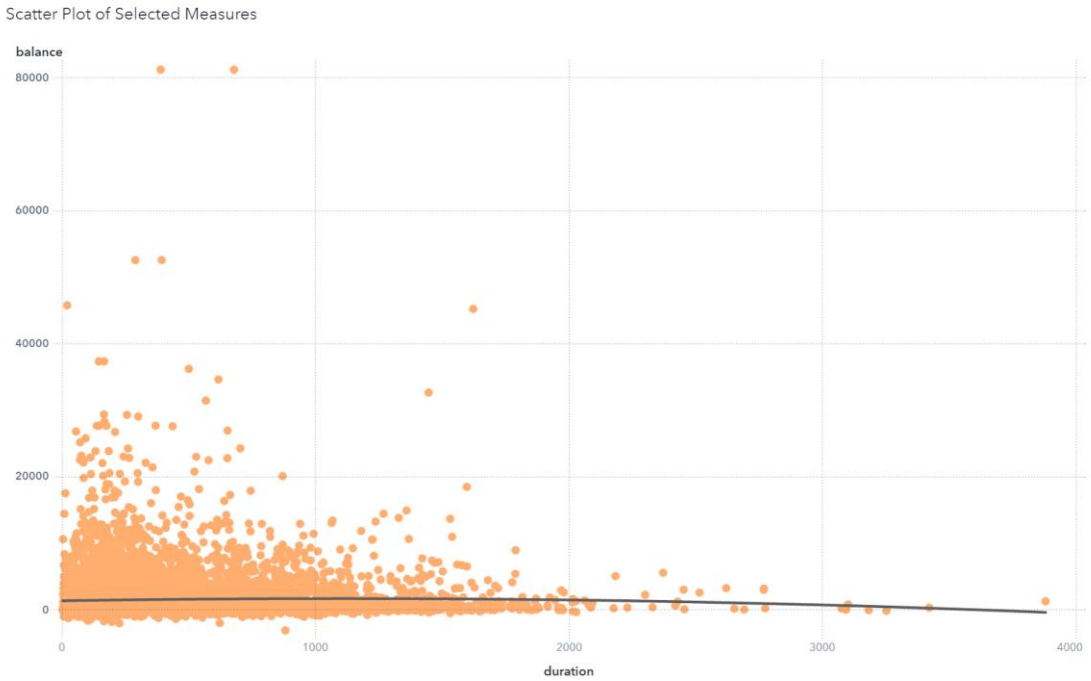


Figure 27: Balance & Duration – Scatter Plot

4.1.4. Campaign and Poutcome

To plot two qualitative attributes together, the parallel coordinates chart was used. In the visualisation, it can be seen that most unknown values of *poutcome* are related to having a value less than 5 in *campaign*. It is also evident that most campaign values fall under 15, and all values higher than 15 result in an unknown value for the poutcome variable. Therefore, it can be deduced that the previous marketing outcome for most target clients who were contacted 5 times or less, is unknown. The same is true for all clients who were contacted more than 15 times. Additionally, as shown below, it is clear that success, other or failure outcomes, do not have an explicit distribution abnormality, and instead are relatively equal. This indicates that other factors identified in previous relationships have a greater impact on the success of previous campaigns.

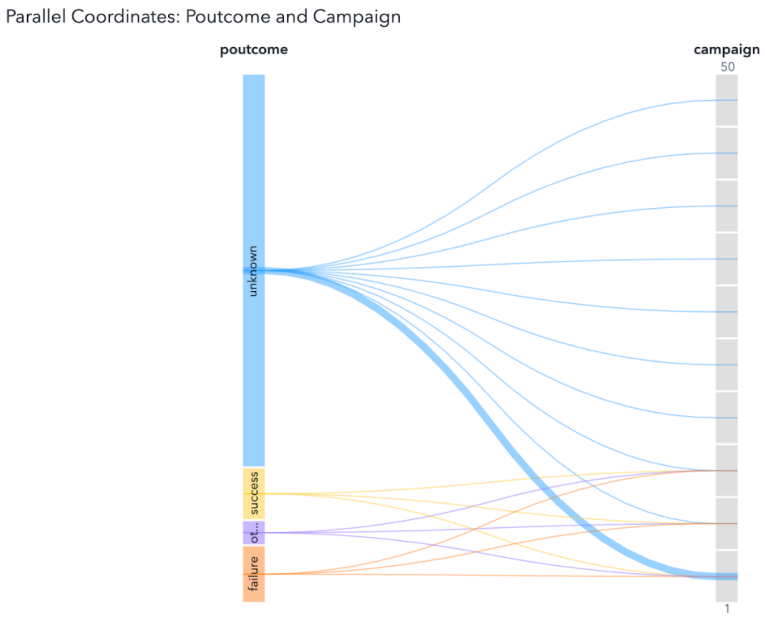


Figure 28: Campaign & Poutcome – Parallel Coordinates Chart

4.1.5. Age and Education

The scatter plot shown in Figure 29 below, indicates that most age groups are from 20 - 80 years old, with a few outliers above 90. Figure 29 also indicated a weak negative correlation between age and education level. A possible thesis for this statement is that the younger generation must undertake mandatory education up to the secondary level in Portugal. The scatter plot reveals that the maximum age for tertiary education is 84. Moreover, only a minority of people in this education group are aged over 80, again reinforcing the potentially new education standards set out for the younger generations.

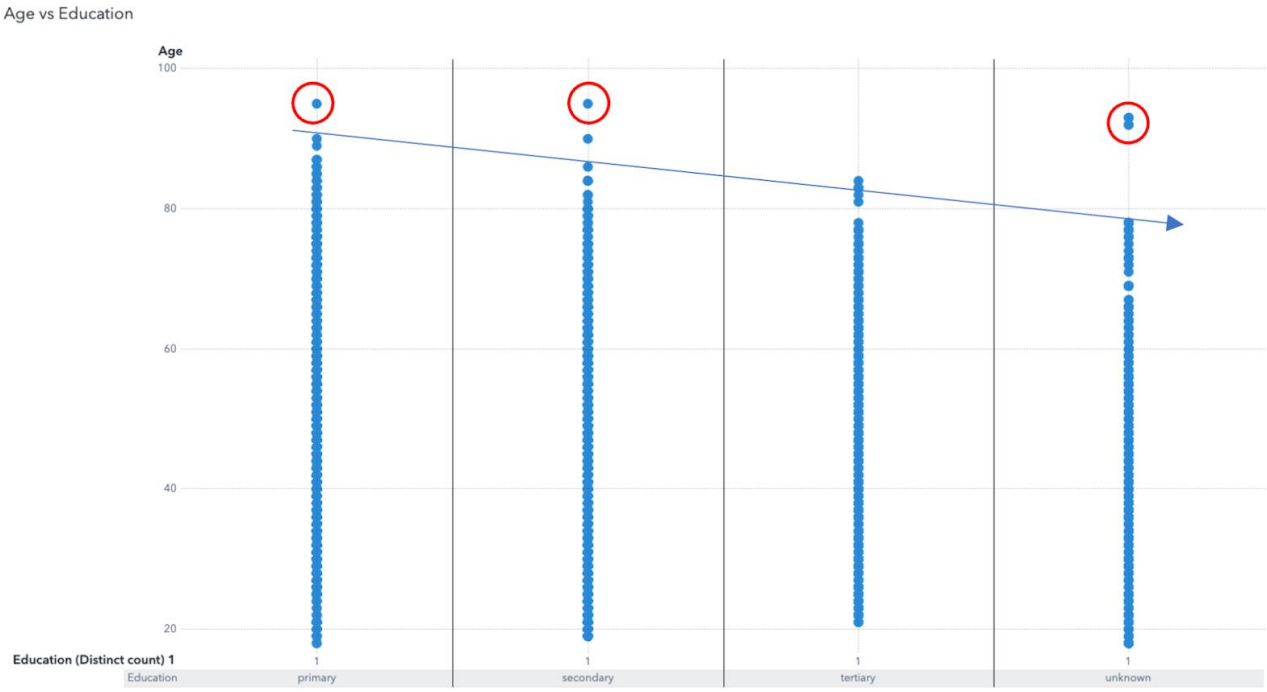


Figure 29: Age & Education – Scatter Plot

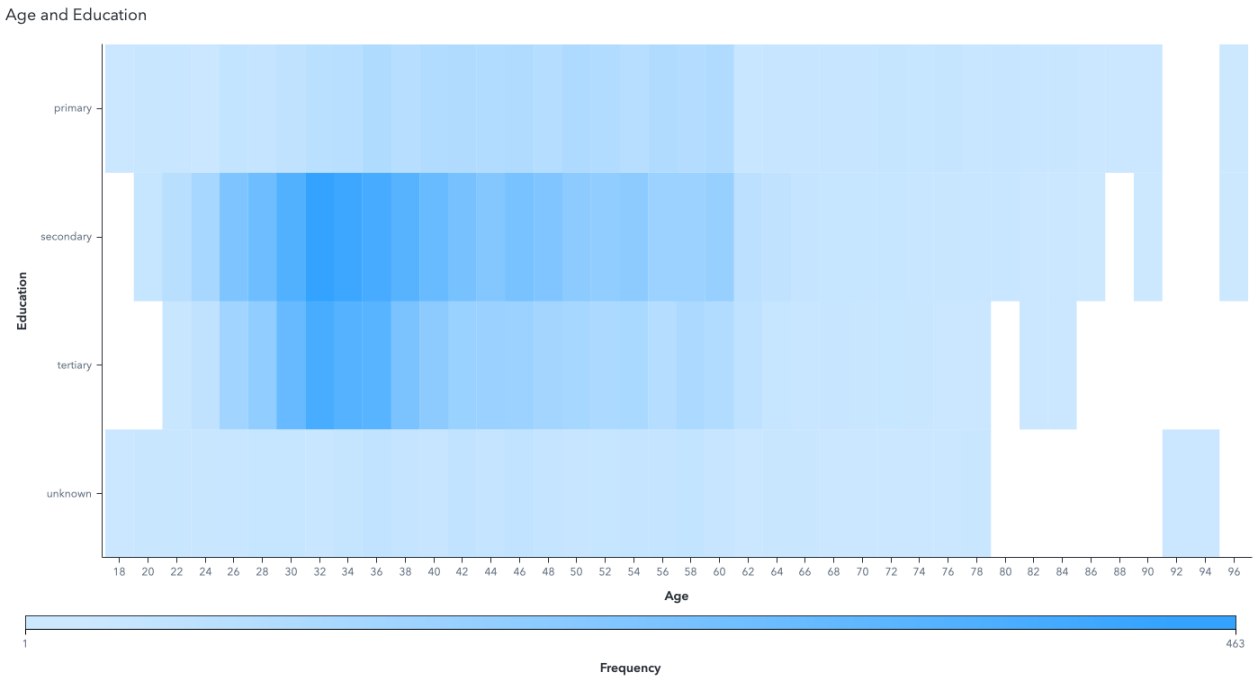


Figure 30: Age & Education – Heat Map

The heat map in Figure 30 also indicates that most people with tertiary education levels are from 26 to 40 years old, with a frequency of approximately 200 - 400 individuals. With a similar frequency, secondary education starts at the same age of 26, but spread out to nearly 60. The distribution of the other two categories is dispersed among all ages with no apparent pattern.

4.1.6. Age and Balance

According to the scatter plot in Figure 31, there is a clear curve, that tapers towards the right due to outliers. Most account balances are below 20,000 euros across all ages, and most ages are in the range of 25 to around 60. This age range is the universal working age range, as well as the approximate working range in Portugal. Target clients below the age of 24, do not have bank balances of greater than 10,000 euros. This is understandable, as most of these individuals may be students or have recently graduated. Potential clients aged 60 or older, have significantly reduced average yearly balances, by up to approximately 50%, to around 10,000 euros. This loose downwards trend creates a weak-moderate inverse relationship, with the increase of age and the decrease of balance. This observation may be due to retirees not earning regular salaries, and instead using their pensions to travel or live a better lifestyle.
The next group of bank balances are from 20,000 - 40,000 euros, potentially representing a group with more stable jobs and higher salaries. Most of this group also is between 25 and 60 years old. The distribution highlights one extreme outlier circled in red, with a balance of 81,204 euros, at the age of 84. This could mean that the individual has a very strong financial situation, or in an unlikely case, there may be a chance the data was incorrectly recorded.

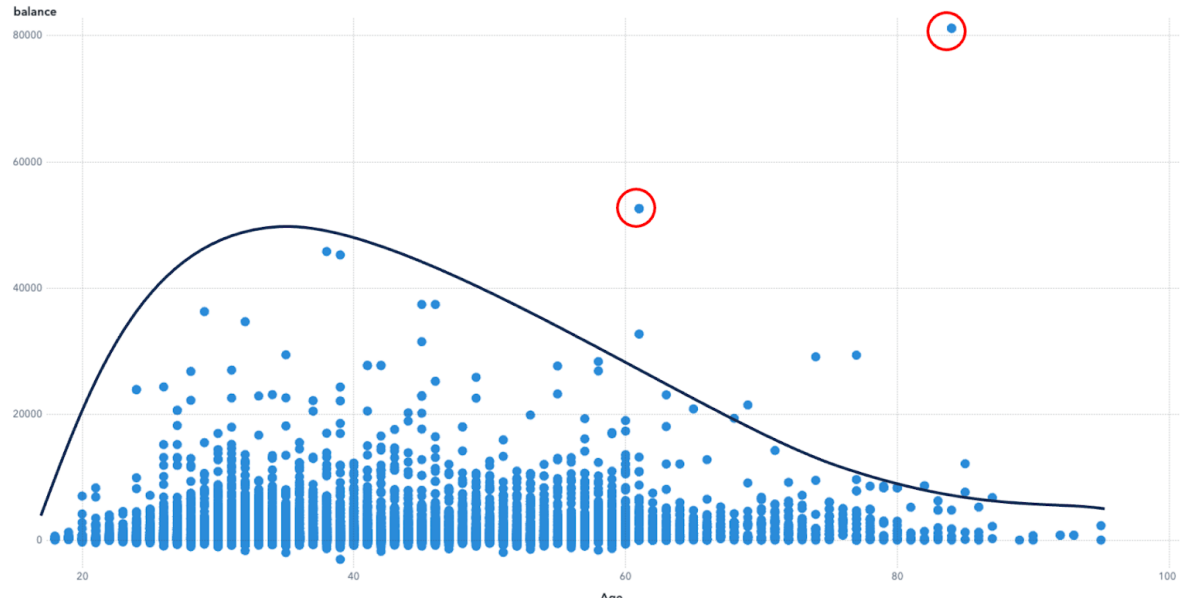


Figure 31: Age & Balance – Scatter Plot

4.2. Multivariate comparisons

4.2.1. Balance, Education and Job

Figure 32 illustrates a multivariate relationship, with job on the x-axis, balance on the y-axis, and education type as the colour of each data point.

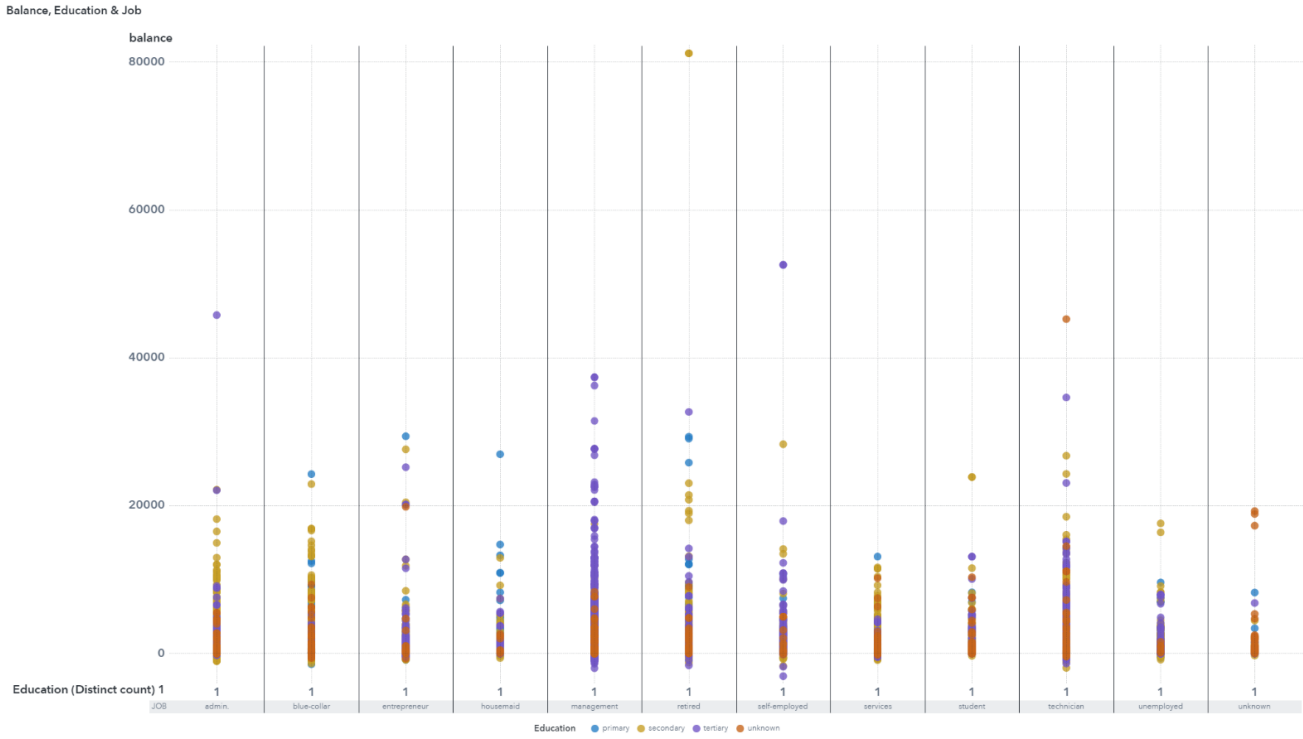


Figure 32: Balance, Education & Job – Scatter Plot

The scatter plot indicates that the majority of people in management roles tend to have the highest bank balances, followed by technicians and blue-collar jobs. This also corroborates the narrative suggested by Figures 3 & 4, where these three roles were the most common in the dataset. This again alludes to the possibility that the bank would prefer clients with stable, high-income jobs, instead of more flexible, but potentially insecure sole-trader roles, such as student or entrepreneur.

Another interesting find is that management roles seem to have higher education levels, with clients completing their tertiary studies. A similar trend is seen in technician roles, albeit to a lesser degree. This suggests that these fields prefer clients with higher education statuses, and therefore attract more competition, and therefore a higher salary. This notion could also justify the bank's disproportionate distribution of data points, in favour of these roles. Contrastingly, whilst blue-collar jobs are also in the top three job types of potential clients, they have minimal tertiary education points, and instead illustrate either unknown or secondary. This coincides with the fact that many blue-collar roles such as electrician, plumber and landscaper do not require a formal tertiary education to provide certification.

4.2.2. Balance, Default and Education

To understand if there is a general trend between education type, personal loan defaults and balance, a scatter plot is used to plot categorical variables education and default on the x-axis, and balance on the y-axis. This multivariate relationship has been depicted in Figure 33 below.

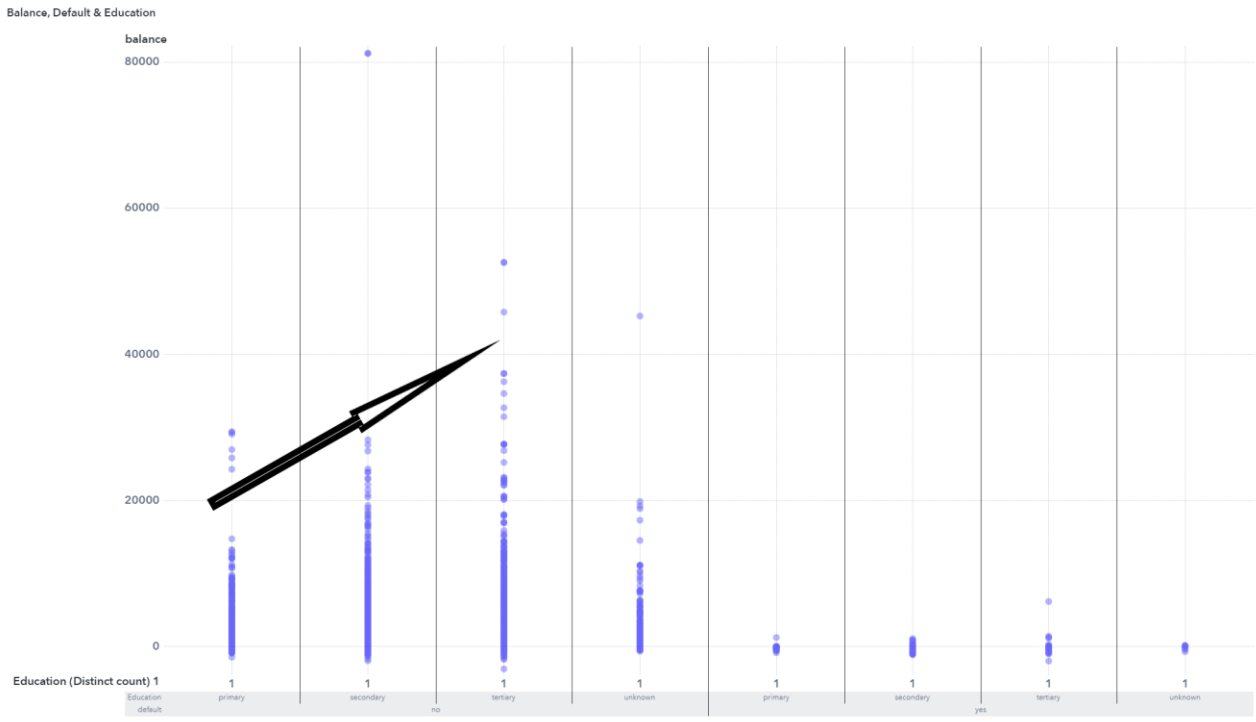


Figure 33: Balance, Default & Education – Scatter Plot

The plot highlights the blatant dichotomy between no and yes default categories, with no being the majority of data points. Within this category, the highest average yearly balance is more closely related to potential clients with tertiary education levels, followed by secondary. This suggests that higher education levels attract higher average salaries. The big difference in 'no' and 'yes' default values proposes that the bank prefers contacting clients that have not defaulted on their personal home loans, and may be more financially flexible. This intentional dichotomy may be strategic on the bank's plan of targeting financially literate clients, who do not like delaying their loan repayments, likely making them loyal customers. Unknown education types that exist within the dataset, may be a resultant of data input error or a lack of recent communication with the potential client.

4.2.3. Age, Balance, Job and Marital

Plotting these attributes in a multivariate scatter plot helps uncover a few weak correlations and trends. In this case, it is evident that the majority of job types do not have their marital status skewed towards a certain type, and instead highlight a relatively proportionate distribution. Two categories diverge from this notion, with 'retired' consisting of predominantly married clients, and 'student' being largely single. This intuitively matches reality, in

which retirees are more likely to be married, whilst students are not. Similarly, the age distribution of retirees is mainly over 60, whilst students are in the 18-50 range. Another interesting finding was that the preponderance of clients who had an average yearly balance of above 20,000 euros, were married. This is not indicative of a strong correlation; however, it could be symbolic of a very weak one. Resultantly, it cannot be said that the bank puts significant effort into telemarketing their term deposit plans, to a higher proportion of married people.

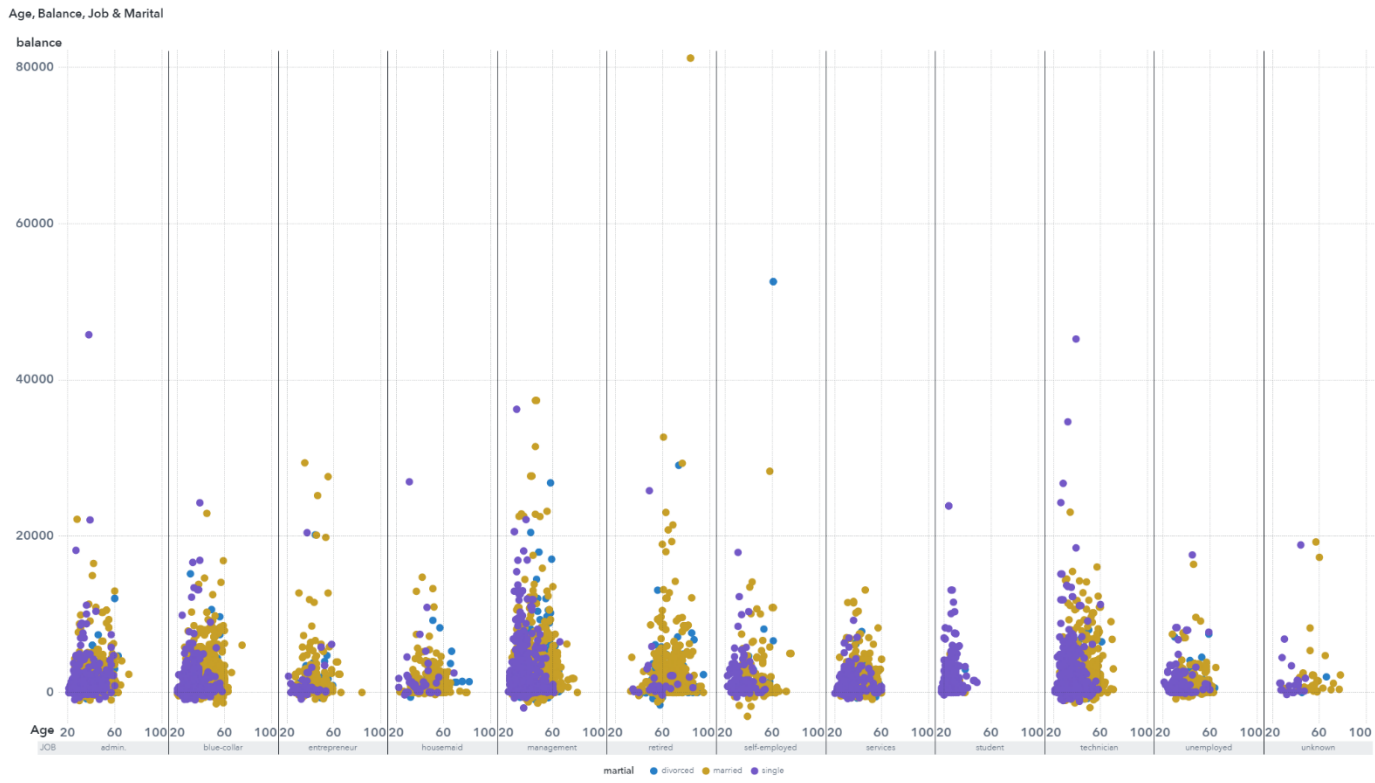


Figure 34: Age, Balance, Job & Marital – Scatter Plot

4.2.4. Balance, Housing and Loan

Figure 35 below, highlights the correlation between balance, personal loans, and housing loans. As shown by the blue dots, potential clients who have no personal loan, also tend to not have any active mortgages. This suggests a weak relationship between both variables. It is also evident that target clients that don't have either type of loan, tend to have higher balances, when compared to those who have an active mortgage. Whilst this an explicit observation, it does not have a strong correlation. This may also be indicative of the fact that mortgages may be more financially troublesome than personal loans, therefore, those with mortgages, have slightly lower average bank balances. The distributions of personal loan categories across those with and without active mortgages is too similar to derive any further insight.

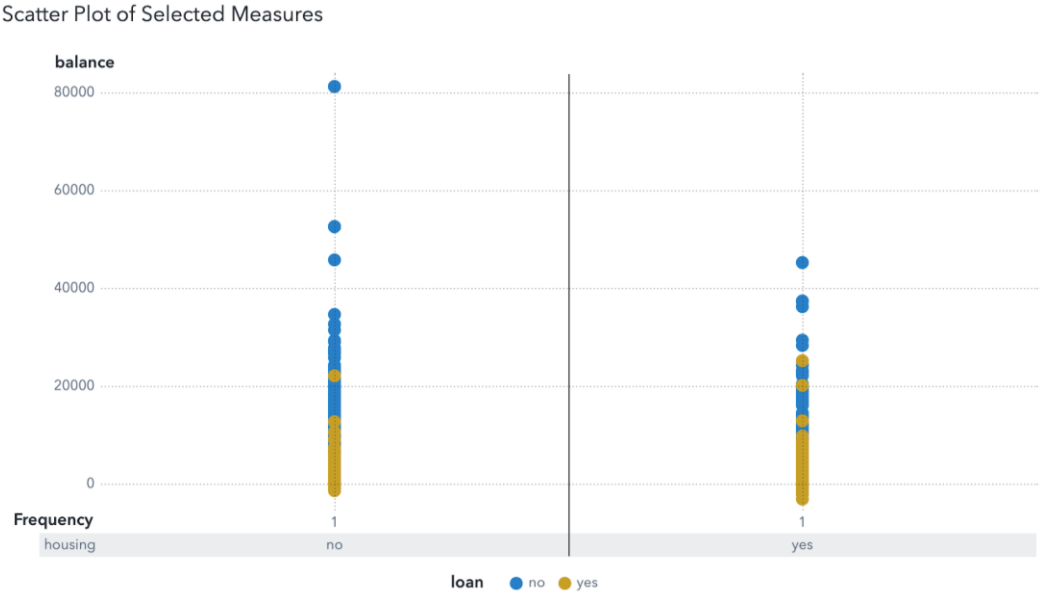


Figure 35: Balance, Housing & Loan – Scatter Plot

4.2.5. Campaign, Contact, Duration and Poutcome

In Figure 36 below, subplots containing '*poutcome*' values of failure, success and other; are grouped towards the left, indicating longer call durations, but fewer contacts during the campaign. The blue points are predominant, suggesting that most contacts during failed, other and success categories, were via cell phones. A few yellow dots can be observed, indicating a lesser reliance on landline communication during these interactions. The purple (unknown) dots, although present, are quite sporadic, showing infrequent instances of unidentified contact methods in these outcome categories.

The unknown subplot stands out both in its size and colour distribution. The most striking feature here is the dominance of purple (unknown) dots, suggesting that campaigns with unidentified outcomes also had a significant portion of interactions with unknown contact methods. This may refer to poor data input templates, not having specific contact options, or data collection errors. Similar with other plots, blue (cellular) dots are still prevalent, indicating that even in cases with uncertain outcomes, cellular communication was prominent. There is a general downward trend, indicating that the higher number of contacts made during the current campaign, the lower the duration of the call.

Across all scatterplots, there's a discernible pattern: data points are majorly concentrated towards the left side, suggesting that irrespective of the campaign's outcome, most interactions were of relatively longer durations, but involved fewer contacts during the campaign (y-axis). Cellular communication remains the preferred mode across all outcomes, highlighting its significance in the bank's communication strategy. Cellular calls also exhibited a generalisation across all plots, with this contact method consisting of the longest durations. This may have been intentional, so that potential clients have a greater likelihood of answering the call. Contrarily, the only exception is the 'unknown' outcome, where unidentified contact methods overshadow other modes. This hints at a potential discrepancy caused by input issues that transcended data collection.

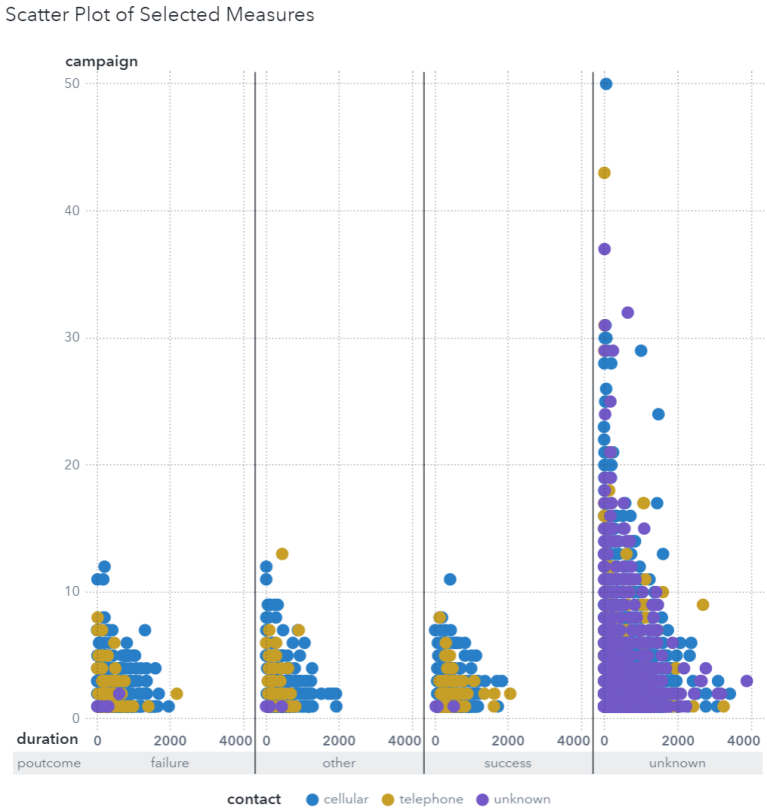


Figure 36: Campaign, Contact, Duration & Poutcome – Scatter Plot

4.2.6. Balance, Pdays and Poutcome

The scatter plot below visualises the relationship between balance, pdays and poutcome. One interesting finding is that all unknown 'poutcome' values are directly connected to PDays, as they all represent target clients which were never contacted in a previous campaign. This indicates that the outcome of the previous campaign is only unknown when the targeted client in that campaign was never contacted. This is a very intriguing discovery, as it provides an explanation for over 70% of the unknown 'poutcome' values present in the dataset.

Another finding is that the clients with higher balances are contacted more recently, particularly between 1-273 days prior to the current campaign. This is dissimilar to the majority of target clients who possessed an average yearly balance of less than 20,000 euros, and were contacted more than 273 days prior to the current campaign. This solidifies the narrative that the bank is more likely to recontact potential clients with higher balances, in the hopes of persuading them to sign up for a term deposit. There are also two main outliers present who have more than 80,000 euros as their average yearly balance. One of these target clients was contacted between 1 – 273 days prior to the current campaign, with the other being contacted more than 273 days prior. Another outlier exists with a 50,000 balance value, albeit significantly smaller.

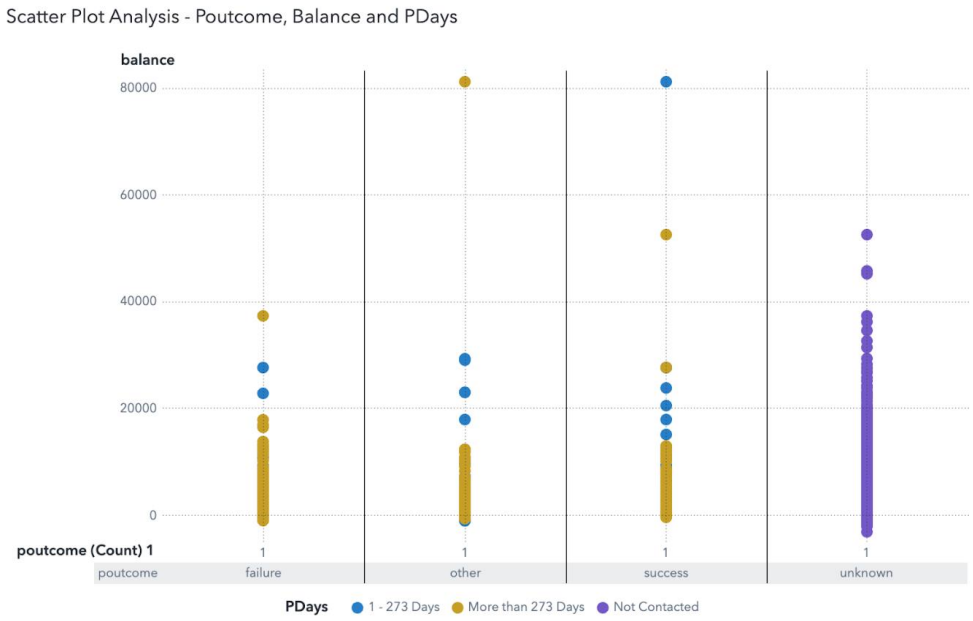


Figure 37: Balance, Pdays & Poutcome – Scatter Plot

4.2.7. Default, Poutcome and Previous

In the scatter plot, it can be seen the almost all unknown values have a 'yes' value for the default attribute. It's also evident that a number of failure values for 'poutcome', also have the default value of yes. A small portion of people of have defaulted on their credit loans, are observed in the 'other', *poutcome* category. In contrast, it is observed that a significant portion of people with a successful 'poutcome' value, have not defaulted on their credit loans. This suggests that those who have not defaulted on credit debt are more likely to have been successful candidates for the previous marketing campaign, indicating that they may have more available cash flow for a term deposit. It can also be seen that all clients who were deemed successful in the previous marketing campaign were contacted less than 25 times. Therefore, it is apparent that too many contacts led to failure or other outcomes for the previous marketing campaign. Interestingly, there is an outlier who was contacted more than 250 times, with an outcome of 'other'.

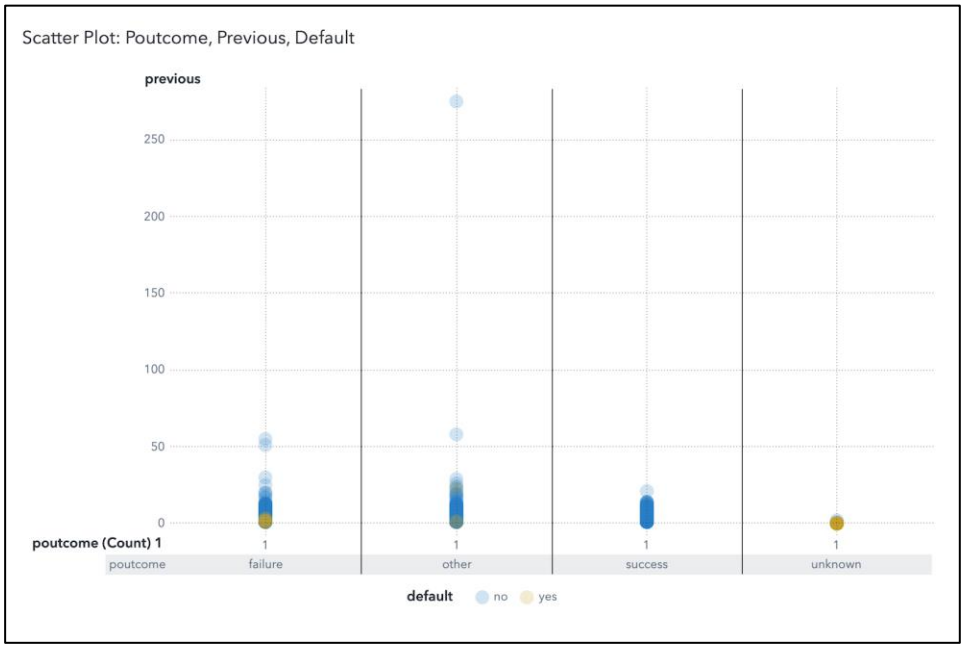


Figure 38: Default, Poutcome & Previous – Scatter Plot

4.2.8. Age, Balance and Loan

Figure 39 indicates that older adults above 60 are less likely to have an existing loan. Only 10 out of more than 10,000 individuals over 60 have a personal loan, as shown in red. This demographic may have a lower risk tolerance, meaning that they are avoiding debt in case unable to repay it from their pensions.

Having a lower bank balance is symbolic of having a personal loan, particularly for the younger age demographic, as explicitly depicted in Figure 40. Making monthly interest payments towards these loans may minimise their bank balance, and motivation to sign up for a term deposit. A subset of the older demographic also has low bank balances, which may indicate their ongoing payments for a personal item, as retirees may not be earning income, and instead are relying on pension payments. Additionally, those with no personal loan commitments are more equally distributed amongst the balance range, and resultantly, no explicit trend can be derived.

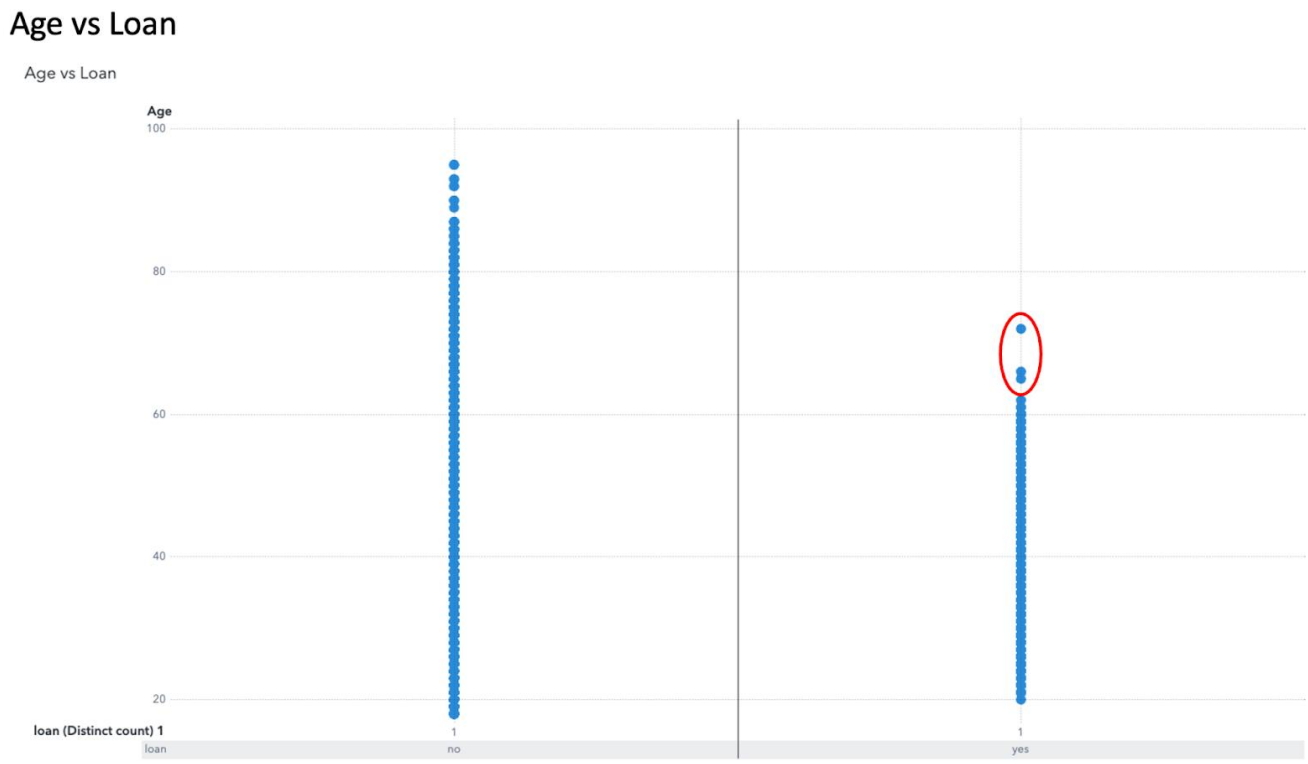


Figure 39: Age & Loan – Scatter Plot

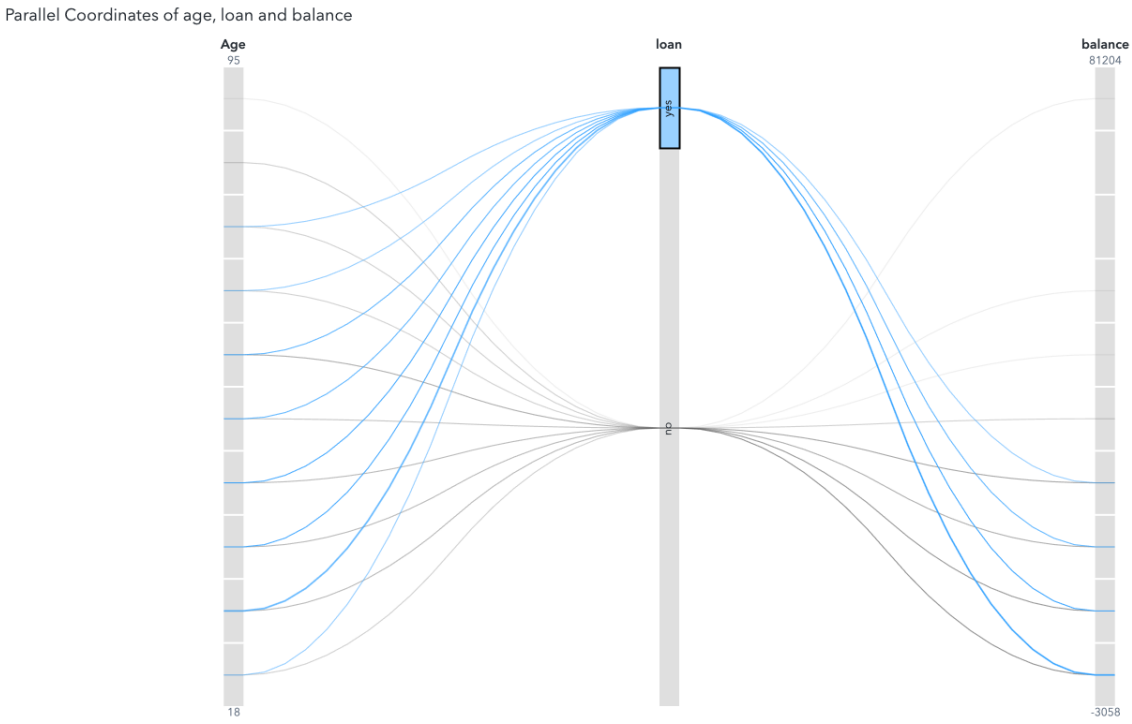


Figure 40: Age, Balance & Loan – Parallel Coordinates Chart

5. Data pre-processing

Data pre-processing involves performing modifications on the input dataset, 'BANK_DIRECT_MARKETING'; so that the data can be more accurately represented based on initial data exploration, and comparative analyses. This process also aims to streamline the dataset by conducting data cleaning, transformation, and dimensionality reduction, so that abnormalities are dealt with prior to predictive modelling.

5.1. Data cleaning

Data cleaning involves ensuring that all records within the BANK_DIRECT_MARKETING dataset are free from corruption, duplication, or omission. As shown in Figure 41 below, there are 24 Age data points missing. Aside from this, the dataset contains no duplicates or corrupt data that requires cleaning.

Obs	Variable Name	Role	Measurement Level	Order	Label	Count	Number of Missing Values
1	Age	INPUT	INTERVAL			75	24
2	Education	INPUT	NOMINAL			4	0
3	JOB	INPUT	NOMINAL			12	0
4	_PartInd_	PARTITION	NOMINAL		Partition Indicator	3	0
5	_dmIndex_	KEY	NOMINAL			254	0
6	balance	INPUT	INTERVAL			254	0
7	campaign	INPUT	INTERVAL			34	0
8	contact	INPUT	NOMINAL			3	0
9	customer_id	REJECTED	INTERVAL			254	0
10	day	INPUT	INTERVAL			31	0
11	default	INPUT	BINARY			2	0
12	duration	INPUT	INTERVAL			254	0
13	housing	INPUT	BINARY			2	0
14	loan	INPUT	BINARY			2	0
15	martial	INPUT	NOMINAL			3	0
16	month	INPUT	ORDINAL			12	0
17	pdays	INPUT	ORDINAL			3	0
18	poutcome	INPUT	NOMINAL			4	0
19	previous	INPUT	INTERVAL			32	0
20	y	TARGET	BINARY			2	0

Figure 41: Missing Value & Attribute register

These 24 Age values have been imputed with the mean for the attribute. This was completed with the imputation node, with a mean value of 41, which was rounded down from 41.2268. This was done to conform with the Age attribute's integer numbering scheme, and can be seen below in Figure 42.

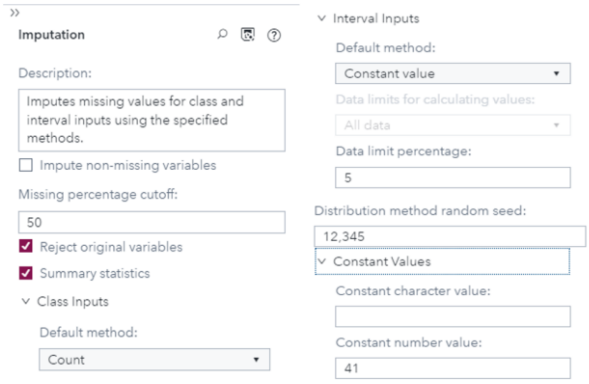


Figure 42: Imputation Node Parameters

5.2. Data transformation

Data transformation involves directly modifying suitable attributes after they have gone through any cleaning processes. Given the lack of data quality abnormalities within the BANK_DIRECT_MARKETING dataset, these transformations will be minimal, and are segregated into binning & data type conversions (SAS, 2021).

5.2.1. Data type conversion

To create a modicum of normalisation across all binary attributes, they were level encoded using a *SAS Code* node to assign numerical and nominal properties to them. These attributes include default, housing, and loan, as shown in Figure 43 below.

```
/* Transform all binary attributes to numeric + nominal */
/* Transformation Method = LEVELENCODE */
Length 'LEVENC_default'n 8;
Label 'LEVENC_default'n = 'Transformed_default_level_encoded';
Length _val_80926280 $5;
Drop _val_80926280;
_val_80926280 = ktrim(put('default'n,$CHAR5.));
if _val_80926280 in ( 'no' ) then
    'LEVENC_default'n = 0 ;
else if _val_80926280 in ( 'yes' ) then
    'LEVENC_default'n = 1 ;

/* Transformation Method = LEVELENCODE */
Length 'LEVENC_housing'n 8;
Label 'LEVENC_housing'n = 'Transformed_housing_level_encoded';
Length _val_80926280 $5;
Drop _val_80926280;
_val_80926280 = ktrim(put('housing'n,$CHAR5.));
if _val_80926280 in ( 'no' ) then
    'LEVENC_housing'n = 0 ;
else if _val_80926280 in ( 'yes' ) then
    'LEVENC_housing'n = 1 ;

/* Transformation Method = LEVELENCODE */
Length 'LEVENC_loan'n 8;
Label 'LEVENC_loan'n = 'Transformed_loan_level_encoded';
Length _val_80926280 $5;
Drop _val_80926280;
_val_80926280 = ktrim(put('loan'n,$CHAR5.));
if _val_80926280 in ( 'no' ) then
    'LEVENC_loan'n = 0 ;
else if _val_80926280 in ( 'yes' ) then
    'LEVENC_loan'n = 1 ;
```

Figure 43: Binary level encoding conversion code

Additionally, both ‘month’ and ‘Pdays’ attributes were changed to an ordinal data type, to better reflect the business problem and data dictionary shown in sections 1 and 3.1, respectively. *Month* was modified to ordinal, to better reflect the standard Gregorian calendar. *Pdays* was changed to ordinal to demonstrate the tiered hierarchy of its durations, as mentioned in section 3.2.15. These modifications may assist in providing more meaningful data during predictions made with neural networks. This may highlight external factors such as the end of financial year, influencing the likelihood of subscribing to a term deposit.

☑	month	Character	Input	Ordinal	Default
☑	pdays	Numeric	Input	Ordinal	Default

Figure 44: Categorical data type modification register

5.2.2. Binning

To perform discretisation of suitable interval attributes in the dataset, it is important to consider binning to reduce absolute values. This has been employed for two main attributes within the dataset: duration and balance. These were chosen specifically to reduce their extremely high-level count of greater than 254, as shown below. To resolve this, equal-width binning was selected, with a total bin count of 15 each. Generally, when using equal-width binning for attributes with high kurtosis values, and platykurtic distributions; outliers may not be accurately represented (LinkedIn, 2023, para. 4). As ‘*balance*’ falls within this description, it is important to combat this without the use of standard bucket binning. Alternatively, tree-based binning was used to optimally segregate values in respect to the target attribute, ‘*y*’; perhaps providing better accuracy, whilst reducing the number of records (SAS, 2018, para. 3; SAS, 2021). As the analysis process can be cyclic in nature, permutations of this transformation method may be employed when determining the best model. A snippet of this binning code is shown in Figure 46 below.

☐	Variable... ↑	Type	Role	Level	Order	Number of Levels
☐	balance	Numeric	Input	Interval	Default	>254
☐	duration	Numeric	Input	Interval	Default	>254

Figure 45: Attributes with high level counts

```
* Transformation Method = TREEBIN ;
Label 'BIN_balance'n = 'Transformed_balance_tree_binned';
Length 'BIN_balance'n $22;
if missing('balance'n) then
'BIN_balance'n= '00:_MISSING_';
else
if 'balance'n <= -888.66 then
'BIN_balance'n = '01:low--888.66';
else
if 'balance'n <= 202.68 then
'BIN_balance'n = '02:-888.66-202.68';
else
if 'balance'n <= 748.35 then
'BIN_balance'n = '03:202.68-748.35';
else
if 'balance'n <= 1839.69 then
'BIN_balance'n = '04:748.35-1839.69';
else
if 'balance'n <= 5113.71 then
'BIN_balance'n = '05:1839.69-5113.71';
else
if 'balance'n <= 5659.38 then
'BIN_balance'n = '06:5113.71-5659.38';
```

Figure 46: Binning code snippet

5.3. Dimensionality reduction

To assist in reducing training time and to remove unnecessary attributes that have no meaning to the value or are highly correlated to each other, it is important to undergo the process of dimensionality reduction. In this dataset, 'customer_id' has been assigned a rejected role (see Figure 47), as it does not provide meaningful data to future predictions, given its sole purpose of record labelling. This assignment will exclude the attribute from any future analysis in the model pipeline.

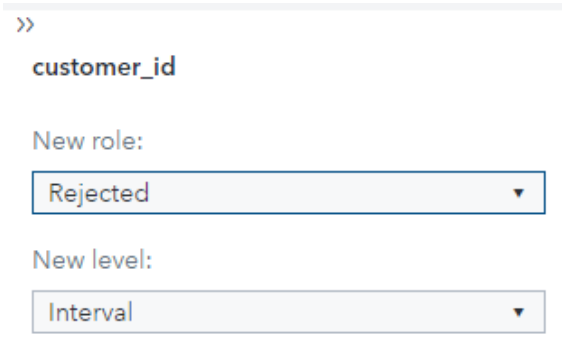


Figure 47: Customer_ID updated role assignment

It is important to note that the dataset has a significant amount of 'unknown' or 'never contacted' (-1) values in attributes such as Poutcome and Pdays. It was originally thought that these unknown values were due to data collection or data input templating errors, as mentioned in section 3.2. Following comparative analysis, it was uncovered that these values were in fact accurate, given the context of the business problem. Unknown and never contacted (-1) values in these variables are related to each other, as they indicate no prior communication, and therefore no prior term deposit registration. A similar notion is exhibited by the 'previous' attribute, where a '0' value denotes that these same target clients were contacted 0 times prior to the current campaign. As is evident, attributes such as these are related, even though the correlation score between them is not extremely high. For example 'previous' and 'Pdays' has a moderate correlation of approximately 41%, as shown in Figure 48. Prior to training, one of these attributes will be omitted to measure the potential dichotomy in model accuracy, when removed.

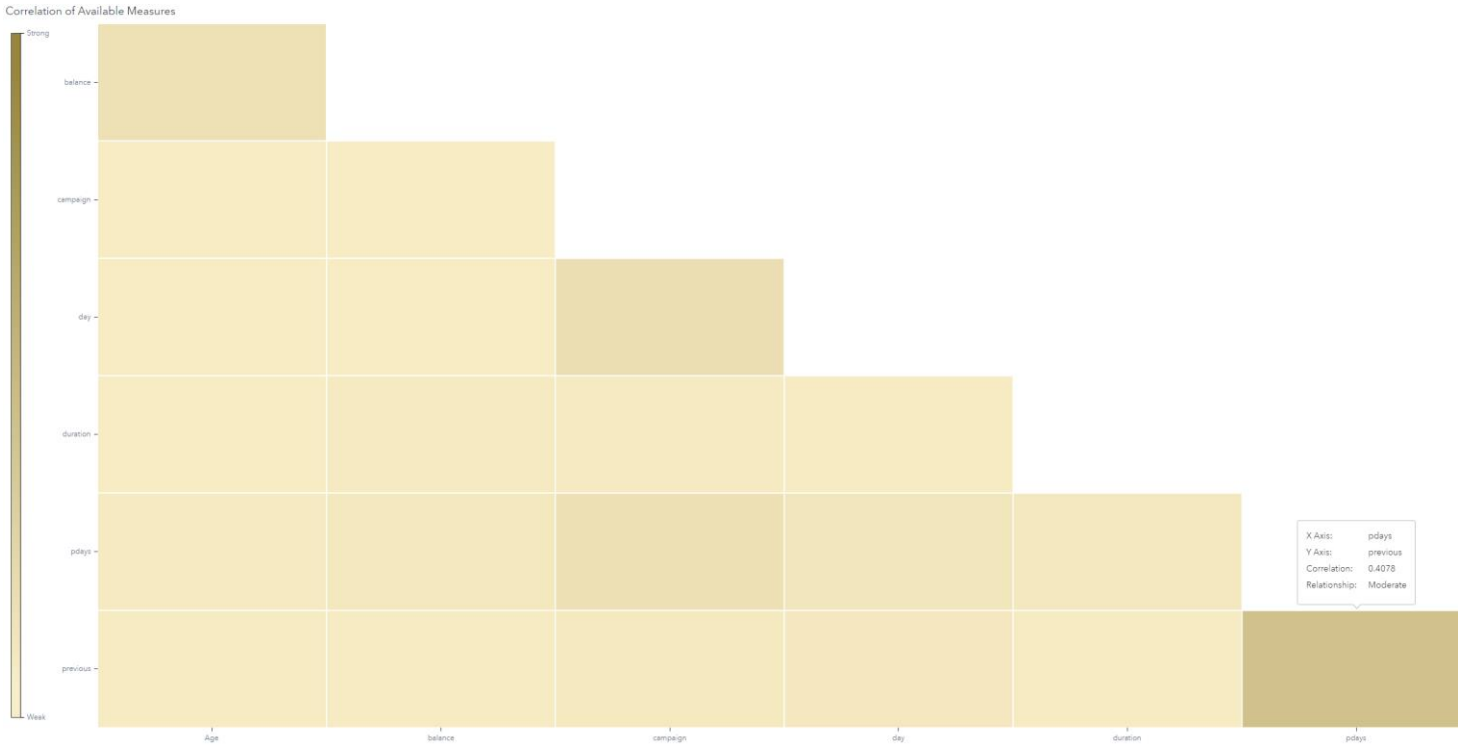


Figure 48: Correlation matrix

5.4. General cluster analysis

To verify that dimensionality reduction steps proposed in the previous section are appropriate for this dataset; clusters, features and correlations, will be examined after pre-processing, to determine if any additional modifications need to occur. Similar to randomised parameter selection for modelling, it is important to note that certain attributes may be omitted to ascertain any accuracy differences in results. Given the nature of this trial-and-error approach, it may act upon decisions made by educated guesses based on the business problem, or randomisation.

5.4.1. Correlation Matrix – SAS Visual Analytics

The correlation matrix shown above in Figure 48, plots compatible numeric attributes against each other in accordance with how closely related they are. Darker shades indicate higher correlations, with the highest being between 'pdays' and 'previous' at 41%. This suggests that both attributes have a relatively higher chance of providing the same insight, when determining if the client subscribes to a term deposit. As mentioned earlier these attributes will be used together and separately, so that any observations in accuracy differences between each combination can be examined. As Figure 48 does not have any highly correlated attributes, no further actions will be taken to omit specific attributes based solely on this matrix.

5.4.2. Relative Importance – Data Exploration Node

The data exploration node in SAS Model Studio utilises a tree-split algorithm to discover the importance variations between attributes in the data set, in respect to the target variable. More precisely, the data exploration node can calculate the relative importance, which denotes the predictive power of each compatible attribute. As shown below, the highest value is 1, belonging to 'duration'. This denotes that *duration* has the greatest influence on whether the customer will agree to a term deposit (*y*). Subsequent attributes such as the outcome of the previous campaign (*poutcome*), and the last contact *month,* both have weak scores of approximately 0.3. The remaining attributes have negligible relative importance scores, and resultantly, these may be used during the trial-and-error attribute omission approach, mentioned in section 5.4.

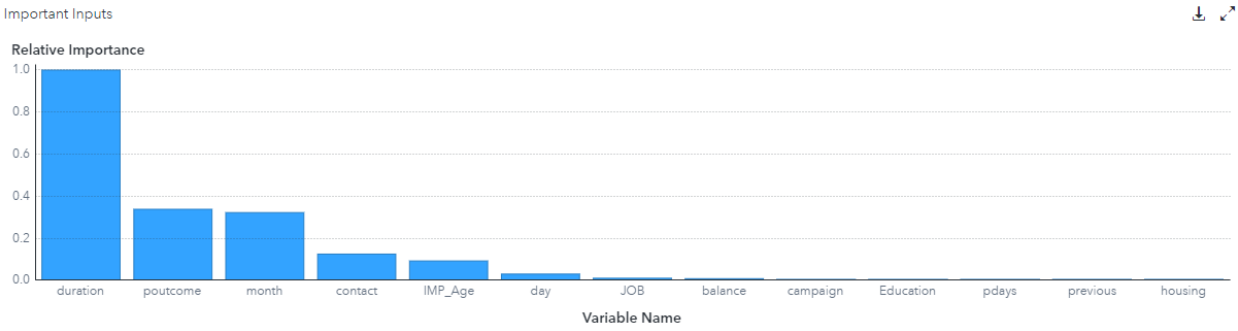


Figure 49: Relative Importance Histogram

5.4.3. Variable Clustering Node

The variable clustering node generates weighted graphs based on computed relation probabilities between various attributes and their subsidiary categories. Figure 50 below illustrates these clusters across all compatible attributes, configured with the use of 25 clustering steps, and an intra-cluster correlation coefficient (RHO value) of 0.8. To allow enough iterations to develop a more insightful cluster diagram, 25 steps was employed, as it sits in the middle of the 50-step limit. Selecting the RHO value is also important, and in this case, 0.8 was chosen. This creates a weighted balance, tilted towards ensuring that two data points randomly selected from within a cluster, are at least 80% more likely to have corresponding values, than two random points selected from the entire set (Sturgis, 2004, para. 7). Evidently, this is a measure of intra-cluster homogeneity, assisting in determining insightful connections (Search.r-project.org, n.d.).

Whilst convoluted, the graph below demonstrates that the majority of all inter-cluster relations are connected to the same attribute, albeit different category values. The greatest weighted connection is between *housing_yes* and *LEVENC_housing*. This is to be expected, as the former indicates if a potential client has confirmed their use of an active housing loan, whilst the latter is a numerical binarisation (completed in section 5.2.1) of the same attribute, i.e. yes = 1 and no = 0. This known relation explains its heavily weighted connection, with a value of 0.517. All other connections do not offer any further insight into inter-attribute relationships, but instead reinforces the general variance between each attribute, potentially aiding in accuracy. This justifies the dimensionality reduction steps carried out earlier, with no explicit modifications derived from this graph.

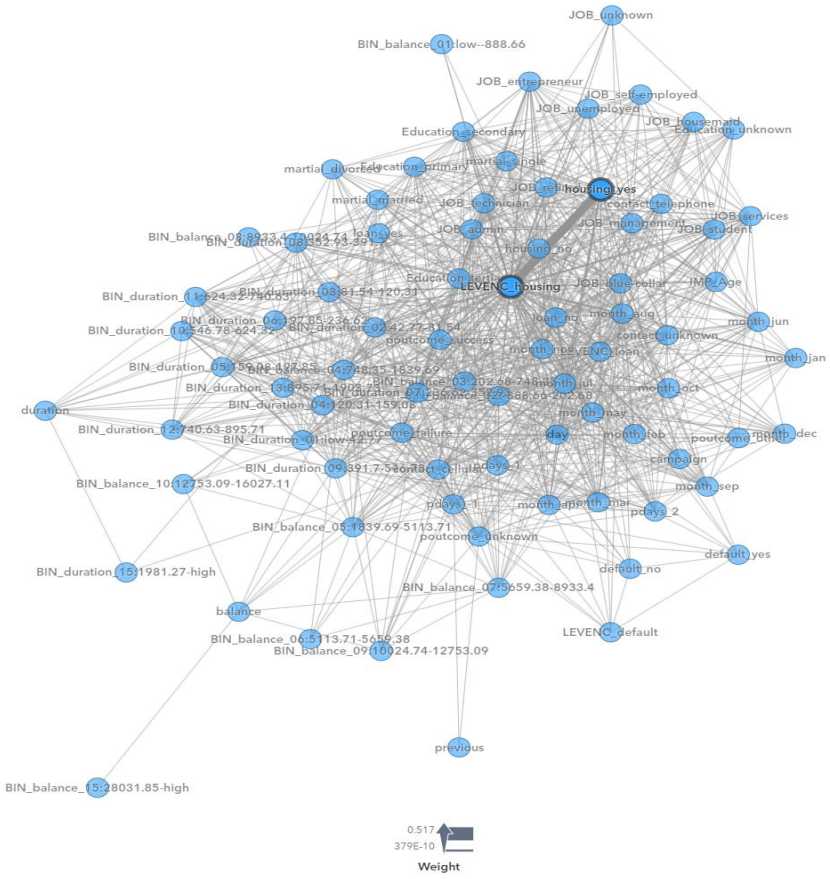


Figure 50: Variable cluster graph with all compatible attributes

To simplify the cluster graph in an attempt to uncover potentially hidden relationships, the penalised log-likelihood cluster selection method was used, as shown in Figure 51 below.

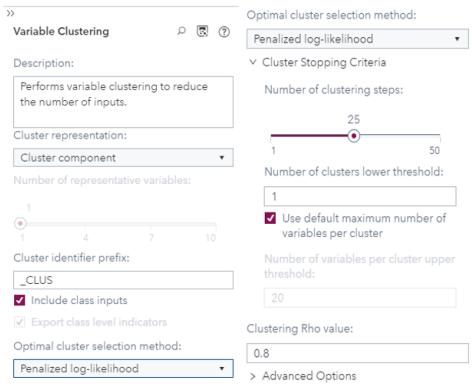


Figure 51: Variable clustering node configuration

The graph below is the output of this new configuration, and it highlights a few key relationships:

- Unknown ‘poutcome’ values have a strong relation to -1 (never contacted) ‘pdays’ values.
    - This connection reinforces the thorough examination conducted in section 5.3 (dimensionality reduction), regarding the same findings.

- Tertiary education has a relatively strong connection with the management job type, and a weak one with blue collar jobs.
    - This relationship also corroborates findings analysed in section 4.2.1 (balance, education, and job), where it was noted that people with management positions were more likely to have completed tertiary education, when compared to blue collar roles.

Other connections in the cluster graph are either negligible and do not provide further insight into the dataset, or are heavily weighted due to being intra-attribute categories.

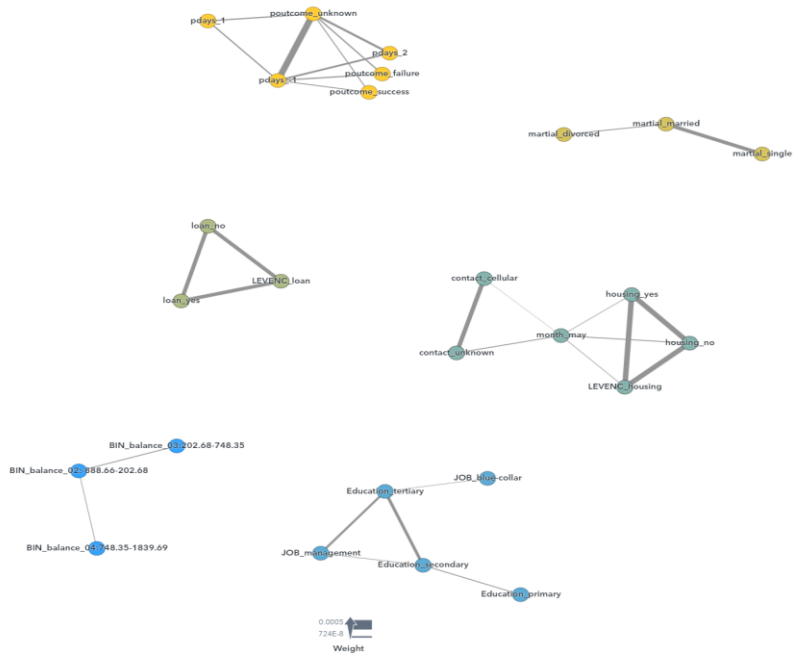


Figure 52: Modified variable cluster graph

5.4.4. Feature Machine Node

The feature machine node is an automated method to attempt at improving model accuracy. It is able to perform complex transformations based to address data quality issues such as high kurtosis, skewness, missing values, and outliers. Examining the details of the results table in Figure 53, it becomes apparent that a myriad of complex transformations have been executed on individual variables, delineating the depth of automated preprocessing

implemented. Noteworthy variables such as *'Transformed_duration_tree_binned'* and *'Transformed_default_level_encoded'* possess relatively high-ranking criteria values, implying their potential prowess in predictive modelling. The high-ranking value verifies that transformations done in section 5.2, prove to be comparatively beneficial as input variables. Moreover, the recommendation of several other transformations such as Box-Cox, median imputation and target / weight-of-evidence encoding may be used in predictive modelling, if accuracy is an issue during modelling.

Generated Features

Obs	Feature	Description	Level	Input Variable	Input Label	Ranking Criterion	Feature Rank
1	cpy_nom_mode_imp_lab_BIN_balance	BIN_balance: Low missing rate - mode imputation + label transformation	NOMINAL	BIN_balance	Transformed_balance_tree_binned	0.01316	1
2	cpy_nom_mode_imp_lab_var_1_	BIN_duration: Low missing rate - mode imputation + label transformation	NOMINAL	BIN_duration	Transformed_duration_tree_binned	0.10258	1

Figure 53: Generated Features – Feature Machine Node

5.4.5. Feature Extraction Node

Feature extraction is a technique that converts high-dimensional data into a smaller collection of features, that maintain the most significant information in the form of principal components. Principal component analysis (PCA) is a technique for generating groups of uncorrelated variables known as principal components, to understand holistic trends across the dataset (Walker & Rogers, 2020). The principal component coefficient represents the weight or the contribution of each original variable to the principal component (PennState Eberly College of Science, n.d.). For example, as shown in Figure 54, principal component 1 correlates most strongly with housing (0.52), and age has the most negative correlation of (-0.49). While this cluster does not exhibit signs of inter-correlation with other generated principal components, this may be indicative of the relatively low dimensionality of the dataset. This can be seen by the drastic differences between components 1 and 2 below. Typically, PCA is conducted to reduce dimensionality with an inordinate number of variables; however, as there is no holistic trend across components, this makes PCA less useful for this dataset. If accuracy issues occur during model prediction stages, this node may be re-examined to determine any overlooked biases.

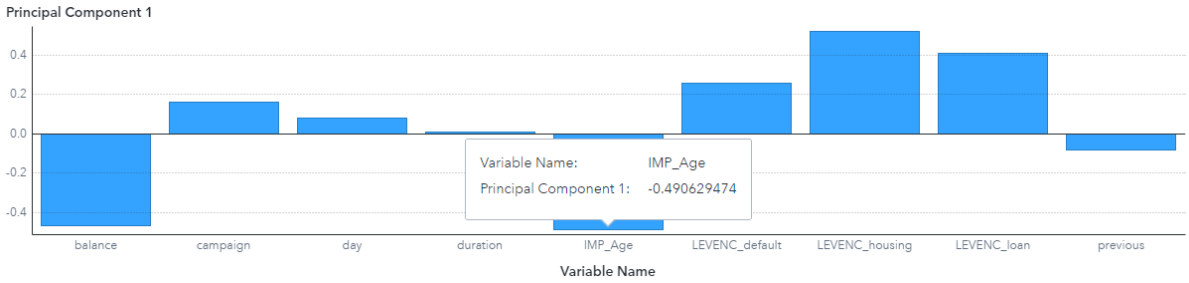


Figure 54: Principal Component 1 – Feature Extraction Node

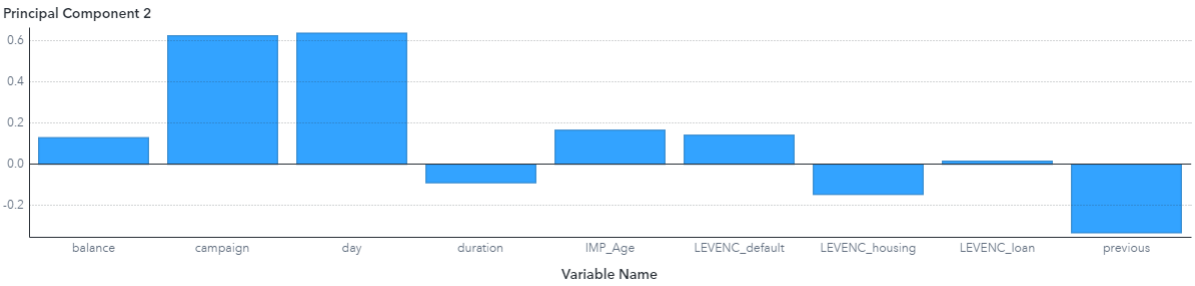


Figure 55: Principal Component 2 – Feature Extraction Node

5.4.6. Anomaly Detection Node

The anomaly detection node is used to detect if outliers are present in the dataset, and attempts to generalise them by dropping observations that may cause any over or under fitting issues during predictive modelling. This node may be used to inform modelling decisions in the next stage of the data analytics cycle.

5.4.7. Clustering Node

The clustering node applies definable automatic pre-processing steps such as imputation, encoding and standardisation, to better segregate data into identifiable clusters. This is more useful with a large dataset with holistic trends identified by principal component analysis; so that smaller training, test, and validation sets can be generated if a subset of the data needs to be examined separately. As this is not characteristic of the BANK_DIRECT_MARKETING dataset, it will not be explicitly used unless specified during predictive modelling.

6. Preprocessing Pipeline

To incorporate the aforementioned preprocessing techniques, the pipeline shown below in Figure 56 will be employed. The pipeline represents the systematic nature of preprocessing used for the BANK_DIRECT_MARKETING dataset, so that classification techniques such as neural networks, random forests and decision trees will have the best chance of achieving high accuracies. These models will be appended to the tree-based binning node and/or the binary attribute transformation node, so that model performance can be compared with and without binning. As specified by the business problem, their purpose will be to accurately determine if a potential client would sign up for a term deposit, by predicting the dependent variable, *'y'*.

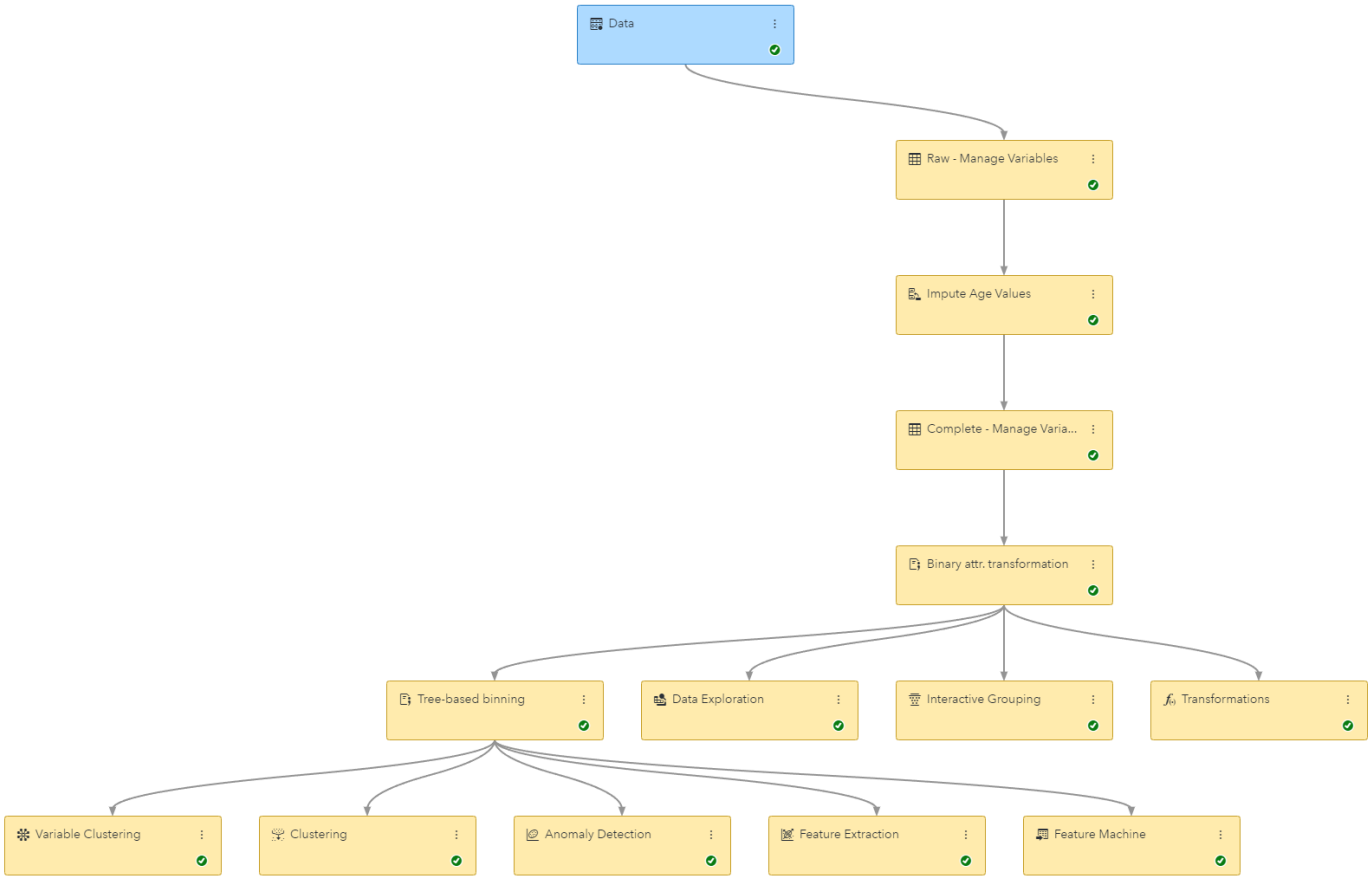


Figure 56: Data exploration and pre-processing pipeline

7. Data mining methodology

7.1. Proposed data mining process

To ensure that a structured testing plan is utilised through model experimentation, it is vital to incorporate a data mining procedure to reduce inconsistent testing techniques and bias within an iterative workflow. The holistic data mining procedure employed throughout this report is visually depicted below in Figure 57.

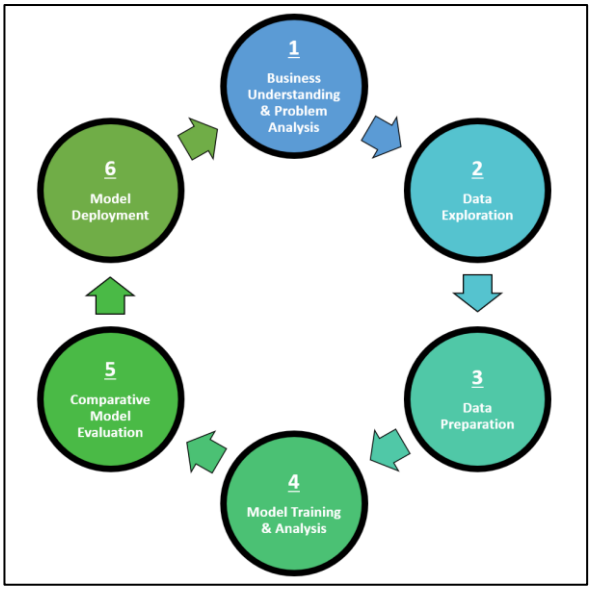


Figure 57: Proposed data mining process

The cyclic diagram shown in Figure 57 illustrates the iterative nature of the holistic mining process used to address the business problem derived from the BANK_DIRECT_MARKETING dataset. Thus far, the report has covered stages one, two and three in their respective sections above. However, to better understand the mining process we have chosen to use, these are described below in relation to the selected business problem of predicting whether a customer signs up for a term deposit, using the given dataset. Abridging the entire process will also explain problem analysis, data preparation and evaluation decisions that have been proposed, so that their potential effects can be better understood and/or justified.

7.1.1. Stage One: Business Understanding & Problem Analysis

Understanding the problem faced by the business is one of the most crucial steps in the mining process, as without a clear and concise purpose, subsequent steps cannot be fulfilled accurately. The BANK_DIRECT_MARKETING dataset in this report contains telemarketing data collected from a Portuguese banking institution, between May 2008 and November 2010 (Moro et al., 2011), equating to 10,578 unique records. By briefly examining input attributes such as education, job type and balance, the business problem becomes prevalent, especially when taking the target variable 'y' into account. This target refers to the inclination of the potential client to sign up for a term deposit, and is therefore a binary attribute. By analysing attribute names, data collection timeline and target variable, it is evident that the banking institution would like to predict if a potential client would sign up for a term deposit. As this dataset was derived from real world telemarketing data, inferences on the institution's motive for the data mining process can be uncovered. These inferences are as follows:

- The banking institution is entering a period of financial instability during the Global Financial Crisis (GFC) in the early-mid 2000's; and would like to predict its ability to gather new long-term cash deposits to support lending functions. Motivated by this change, it may be in the bank's best interest to contact clients from different demographics who have yet to make any term deposits.

- The banking institution would like to determine if specific demographics such as job types, education or previous campaign outcome impact the client's inclination to sign up for a term deposit. This is something of great importance to them as they can narrow down their marketing efforts to target clients that are more likely to sign up. This notion is further accentuated given the external factors caused by the GFC during the data collection time period.

Based on this analysis, it was defined that an accuracy goal of at least 75-80% (Hendricks, n.d.; Parashar, 2023), on the test set would be our aim, for this binary classification problem. To gain a broader understanding of model performance, accuracy will not be used as a sole performance metric for comparative analysis, during the later stages of the data mining process. Instead, it may be supplemented with other metrics such as F1 scores, KS coefficients, Average Squared Error (ASE) and Area Under Curve (AUC). To ensure fair testing, another goal was derived from the problem analysis; ensuring that during the evaluation stage, models would be compared systematically, while being metric agnostic. This can assist with effectively eliminating worse performing classifiers,

without overlooking strengths found in other iterations, if indicated by supplemental performance metrics such as KS or F1 scores (Google for Developers, 2022).

7.1.2. Stage Two: Data Exploration

Analysing the banking institution's dataset for patterns, preliminary insights and distribution statistics is an essential step during the data mining process, and has been thoroughly completed in section 3 above. This analysis enabled key insights to be formed such as the discovery of management roles intrinsically preferring tertiary education. This demographic contributed to a major portion of the dataset (approximately 22%), alluding to the narrative that these stable roles attract higher average yearly balances; and therefore may be more willing to subscribe to a term deposit.

One prominent insight into the dataset was uncovered after thoroughly examining attributes referring to prior interactions from different marketing campaigns. It was found that there was a strong connection between Pdays and Poutcome attributes, where 'unknown' Poutcome values, indicate that the target client was never contacted before. This finding helped explain the enormous disproportionate distribution of 74.4% of all records being marked as 'unknown' for the Poutcome attribute. It was unveiled that all unknown values were directly linked to clients who were never contacted before ('-1'), as described by the Pdays attribute. This creates a strong holistic narrative that the Portuguese banking institution was skewed towards targeting a majority of new clients, instead of previously contacted clients. This decision could have been motivated by the global financial crisis at the time of data collection; and executed as a bid to increase their term deposit signup rate by introducing new targets. This significant finding corroborates the inferences derived from section 7.1.1 above, where new target clients may be of greater interest, given their higher likelihood of having free cash flow; in relation to repeat clients who may not have as much disposable income. Additionally, it was also ascertained that the dataset was well prepared and collected, with very few missing values equating to a mere 24 out of all 10,578 records.

Attributes such as 'Loan' exhibited a clear skew towards the 'no' category (87.1% of dataset), indicating that the majority of potential clients approached by the bank's telemarketing campaigns, did not have other personal loan commitments, potentially making them more financially flexible. This again reinforces the bank's likely intention of targeting a new demographic for their latest round of telemarketing campaigns. This theme is noticed throughout many bi and multivariate relationships, and provides insight into the rationality of the bank's marketing decisions. However, while many individual variable distributions were skewed towards a singular category as mentioned above, examining the overall dataset's characteristics proves to be otherwise. The overall dataset is perfectly balanced, with 50% of 'yes' and 50% of 'no' values present in the target variable, 'y'. The well-balanced nature of the dataset could make data preparation much simpler, as Synthetic Minority Oversampling Techniques (SMOTE) would not be necessary, and would instead add extra pre-processing overhead in the pipeline, without much benefit (Dang et al., 2013, p. 237). This characteristic would also enable easy-to-interpret performance metrics such as accuracy to be used without misrepresenting the actual distribution of the dependent variable, 'y'. Moreover, within the BANK_DIRECT_MARKETING dataset, an 80% accuracy value is not representing an underlying distribution where the dataset itself contains 80% 'yes' values, and 20% 'no' values. If this were the case, it cannot be said for certain that the model is able to accurately predict 80% of new records; instead it will be more likely that new data will be misclassified as 'yes', due to the initial data imbalance during training (Zach, 2021). As this is not an issue in the dataset, accuracy statistics during model training and comparative model evaluation, would not pose any explicit biases on new data, and would suffice for a suitable relative comparison metric between classifiers (Zach, 2022).

7.1.3. Stage Three: Data Preparation

Data preparation is a key step in ensuring that predictive models have the best possible chance at attaining a high classification accuracy. Depending on the dataset, this stage can be more or less crucial to the success of trained models, depending on how convoluted the input data is, and if any biases or redundant attributes exist. Ultimately, this stage was chosen to be completed prior to model training so that data can be more accurately represented based on initial data exploration, and comparative analyses. This process also aims to streamline the dataset by conducting data cleaning, transformation, and dimensionality reduction, so that abnormalities are dealt with prior to predictive modelling. As gleaned during data exploration, the BANK_DIRECT_MARKETING dataset did not have major class imbalances or overall skewness issues. However, to address the small number of 24 missing age values, integer imputation was executed, using the mean age of 41. To create a modicum of normalisation, all binary attributes such as default, loan and housing were also level encoded to assign nominal properties to them, whilst being consistent with numerical labels.

As is evident, these changes are minor in nature, given the cleanliness of the input dataset; however add to overall readability and consistency of the pre-processed set, ensuring clean prediction results. Similarly, 'month' and 'Pdays' were modified to an ordinal data type to better reflect the business problem explained earlier, aligning the 'month' attribute with the Gregorian calendar, and structuring 'Pdays' in accordance with its graded hierarchy of durations. These changes were executed with foresight, knowing that neural networks may use the ordered structure of the attributes to provide more accurate predictions, potentially highlighting external factors such as the end of financial year, influencing the likelihood of subscribing to a term deposit. Changes like these again reinforce the structure of our data mining process, highlighting the importance of undertaking preparation changes prior to training, for a possible accuracy gain.

During the data exploration stage, it was observed that account 'balance' and 'duration' had extremely high-level counts of greater than 254, given their naturally discrete nature; and exhibited high kurtosis values and platykurtic distributions. Attributes with these characteristics are prone to inaccurate outlier representations due to their long tails. To combat this, tree-based binning was utilised for 'balance' and 'duration' attributes as they fall into this category, with 'balance' exhibiting a greater kurtosis value of 119.6, compared to 7.4 for 'duration'. This binning methodology utilises a tree split procedure to optimally segregate values in accordance with the target variable 'y', with a chosen value of 15 to avoid over-sparsification.

It is important to note that the dataset has a significant amount of 'unknown' or 'never contacted' (-1) values in attributes such as Poutcome and Pdays. It was originally thought that these unknown values were due to data collection or input errors. Following the data exploration stage of the mining process, it was uncovered that these values were in fact accurate, given the context of the business problem. Unknown and never contacted (-1) values in these variables are related to each other, as they indicate no prior communication, and therefore no prior term deposit. A similar notion is exhibited by the 'previous' attribute, where a '0' value denotes that these same target clients were contacted 0 times prior to the current campaign. As is evident, attributes such as these are related, even though the correlation score between them is not extremely high. For example 'previous' and 'Pdays' variables have a moderate correlation of approximately 41%. Identification attributes such as 'customer_id' were also omitted from the dataset prior to performing training.

As the training and preparation stages may be cyclic as shown in Figure 57, permutations or modifications of the aforementioned preparation methods may be employed when determining the best model.

7.1.4. Stage Four: Model Training & Analysis

Model training was intentionally chosen to occur directly after data preparation so the input data does not undergo any further changes that may detriment the final accuracy value, without explicit documentation. To attempt at achieving an accurate model to predict customer term deposit sign ups, it is important to utilise various machine learning classification techniques. To align with this, the advanced classification pipeline template was used within SAS Model Studio, containing neural network (NN), logistic regression (LR), decision tree (DT), gradient boosted tree (GBT), and random forest (RF) classifiers. A support vector machine (SVM) classifier was also added to this lineup to create a more robust evaluation procedure, using all available classification models present in SAS Model Studio. Linear regression was omitted from this collection, as it is primarily used with continuous targets, instead of categorical ones, as is in this dataset. Whilst linear regression could be implemented with the use of level encoding as carried out on 'housing', 'default' and 'loan' variables; due to SAS limitations, target variables ('y') cannot be modified after pipeline creation. Additionally, incorporating linear regression would require a threshold value to be set as predictions will be in the form of continuous decimals from 0 and 1. As the function is linear, with the addition of a new data point, the line must rise or lower to re-generate the line of best fit for the newly added point. The issue with this method is that the original threshold value will be shifted at each data point, and may end up at a completely different value, from the standard 0.5. Evidently, this taints the prediction accuracy, which would limit the insight given from a linear regression model to the Portuguese banking institution. This same characteristic is not present in logistic regression models, as they utilise a sigmoid function instead (Kumar, 2021).

During our model training process, each classifier will go through at least 10 training iterations, with different parameters, to attempt at boosting individual performance metrics. To aid in intra-model comparisons between iterations of the same classifier, a standard set of baseline performance metrics will be utilised to make relative comparisons straightforward. These metrics include accuracy, F1 scores (Prabhakaran, n.d.), KS coefficients, Average Squared Error (ASE) and Area Under Curve (AUC). Whilst this selection of metrics may aid in the selection of the best iteration per classifier, KS scores will be used to select the highest achieving model for each classifier, supplemented

by accuracy to determine the KS score's reliability or validity. This is important, as within unbalanced datasets, the distribution of the target variable is not equally divided across yes and no categories. If a completely unbalanced training set were to exist, a YES value would be converted to NO, and the same goes for NO to YES. In this case, the separation between the KS cumulative distribution functions (CDF lines) for both YES and NO would achieve a maximum height as the classifier would be able to separate classes almost perfectly with an extremely high KS value. This notion may also indicate severe overfitting, and can be seen in Figure 58 below, in the perfect classifier chart. However, when conducting model testing it would be obvious that the model does not perform as intended, as the accuracy score will be very low given the inverse target variable. This is why we have chosen KS scores to be used in conjunction with accuracy to act as the primary determinant for intra-model comparisons during iterations of a single classifier. As mentioned earlier, the target variable 'y' in the dataset is balanced evenly, so this will not be an issue, however the accuracy score will be used to reinforce the KS score; with relatively high KS scores, equating to relatively high accuracies. Holistically, it is our prediction that ranking KS scores from high to low, would also in effect, **generally** rank models by accuracy (high to low), as they are intrinsically related (Trevisan, 2022; Cross Validated, n.d.).

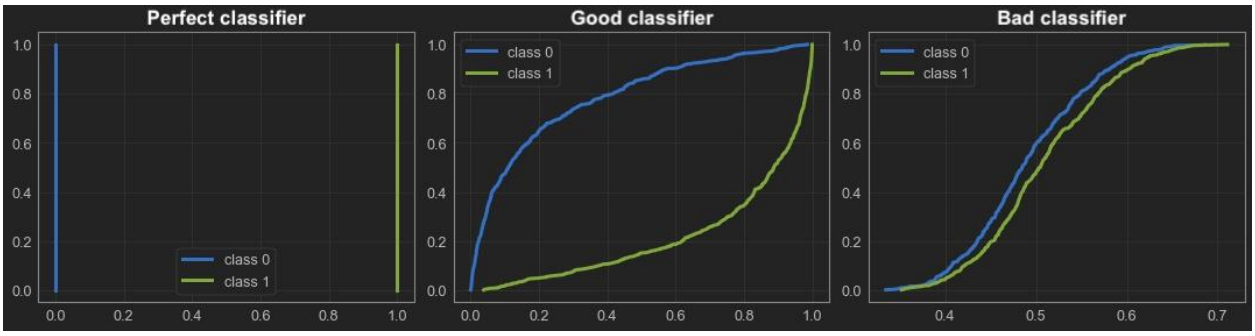


Figure 58: KS Value Charts (Trevisan, 2022)

It is also vital to note that for unbiased model training and analysis, a constant random seed (12345) will be used for all classifier iterations, so that the same subset of data is used for training, validation, and test datasets. In certain classifiers such as decision trees, this may also mean that randomised rule generation and node pruning penalties will be reproducible, and therefore equitably comparable between iterations. By adhering to this rule, model performance metrics can be evaluated fairly, without a potential for accuracy gains due to the distribution of the randomised data split itself (SAS Help Center, n.d.; Arora, 2022).

7.1.5. Stage Five: Comparative Model Evaluation

Similar to intra-model comparisons, this stage compares all winner models with each other from each classifier. Neural Network, Logistic Regression (Forward, Backward & Stepwise), Random Forest, Decision Tree, Gradient Boosted Tree, and Support Vector Machine models will undergo comparative analyses using the following performance metrics:

- KS Youden Statistic:
    - The KS performance metric evaluates models by calculating their maximum or highest point of difference between their negative and positive curves - meaning that a higher KS score (0-1) suggests that the model is better at separating predicted class probabilities from each other (Prabhakaran, n.d.; Khan, 2017). Therefore, it is a measure of prediction certainty, with higher KS values referring to the model's high probability of predicting if a customer signs up for a term deposit.

- Gini Coefficient:
    - The Gini coefficient is an adaptation on the AUC subsidiary component described below. It transforms AUC to a more readable range, adding 0 for a random model, -1 for a worse model, and 1 for a perfect model. Adding a negative range to the standard AUC metric makes it clearer to interpret in some classifiers (K2 Analytics, 2020). Subsidiary components used to calculate the Gini coefficient are described below in relation to the BANK_DIRECT_MARKETING dataset:
        - Receiver Operator Statistic Curve (ROC) - subsidiary component:
            - The ROC graphs are a method used to plot all threshold points available during classification tasks. For example a 0.5 threshold would refer to values with a probability of greater than 0.5, as a predicted customer term deposit signup. ROC graphs plot the true positive rate (TPR) along the y-axis, and the false positive rate

(FPR) along the x-axis. As circled in green in Figure 59 below, it is evident that ideal thresholds would be towards the upper left corner for this business problem, as true positives are of more significance than compromising by adding false positives, just to achieve a higher TPR (StatQuest, 2019). This so that the Portuguese banking institution can correctly forecast their available cash flow, without risking the addition of false positives, which may lead to the employment of extra staff to handle future telemarketing calls, when they are not needed. Figure 59 below illustrates this notion, with reference to ROC, AUC, TPR, FPR and Gini.

- Area Under ROC Curve (AUC) - subsidiary component:
  - AUC metrics assist in comparing two or more different ROC graphs to each other by calculating the area beneath them. A classifier with a higher AUC value, predicts more targets correctly, with better probabilistic differentiation between a successful or unsuccessful client term deposit registration (MLNerds, 2021). Therefore, AUC will be used for comparative analysis either solely or as a part of the Gini coefficient, as it is a strong relative measure, in a single value for all thresholds. This will be particularly useful for final inter-model comparisons.

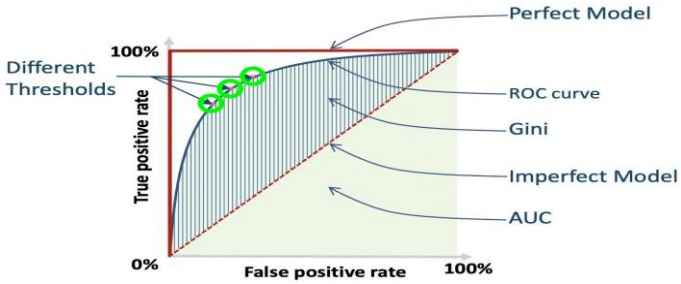


Figure 59: Ideal ROC Curve (Khal, 2021)

- Accuracy:
  - This is a measure of correct predictions out of total predictions, and can be used for inter-model comparisons, due to having a perfectly balanced dataset.

- Average Squared Error (ASE):
  - ASE is another relative measure of how well a model is able to generalise, without an inordinate number of outlier predictions (higher values indicate worse performance). Due to the square function, these errors will become more obvious. Contrastingly, if a classifier makes a single bad prediction on a customer term deposit signup, this will be exemplified due to the square function, so this metric will not be deterministically used, but instead used as a supplement to other performance metrics mentioned above (Seif, 2019).

- Schwarz's Bayesian Criterion (SBC):
  - The SB criterion is helpful in determining which the most efficient model out of a finite set of trained classifiers. It utilises a likelihood function, taking into consideration the number of input attributes required to achieve a certain accuracy score for a particular model (Devansh, 2021). It then ranks and estimates all models based on this, providing penalties for models who use more attributes, yet achieve the same or lower accuracy scores (or higher errors). This is mainly employed to reduce dimensionality further within an iterative trial-and-error attribute omission loop, and therefore cannot be utilised effectively given SAS Model Studio limitations. However, manual inter-model comparison using this statistic may be used to detect any space and time efficiency gains between certain classifiers such as stepwise and forward logistic regression, with lower values indicating better models.

Following the successful determination of the best performing model (champion), the model comparison node will be used to deploy and score the champion model, along with the challenger (2nd best model).

7.1.6. Stage Six: Model Deployment

Model deployment is a key step in making the best model available for use with real world or live data. This enables various products to be developed using predictions made by deployed models, which run in the cloud. SAS Model

Manager enables champion models to be implemented in the cloud to begin new data ingestion and prediction. SAS also enables the use of periodic champion-challenger comparisons to occur, so that continuous model improvement can occur over time, highlighting the cyclic data mining process. Towards the end of the report, the champion model will be implemented within SAS Model Manager to conduct a scoring test prior to deployment.

8. Modelling techniques & experiment analysis

8.1. Neural Network

The neural network model aims to artificially mimic human brain cells in a black box environment, assisting in producing certain outputs based on specific inputs. The neural network architecture is popular for its ability to process complex and high dimensional data, with less data being prone to overfitting. Between the input and output layers, there are many hidden layers containing activation functions aiming to find any possible relationship between the input and output, by adding relevant weights as data gets fed through the network. As neural networks are mostly dataset agnostic, they can be used in many tasks, including binary classification. For this reason, it was chosen for the BANK_DIRECT_MARKETING dataset, as explicit preprocessing was unnecessary. Neural networks can also identify hidden features, directing more resources to analyse those critical features to enhance model performance. To assist in this, section 5.2.1 highlights data type changes that may provide further information to the neural network, based on the order of 'month' and 'pdays' attributes.

Whilst the interpretability of neural networks is low, its ability to ingest large quantities of data, makes it one of the most popular classification techniques. However, as the BANK_DIRECT_MARKETING dataset is only 10,578 records long, it may not achieve the full potential of the model, as neural networks prefer data an order of magnitude greater to generate a robust working model, if extremely high test set accuracy is desired.

Several factors can influence the result of neural networks, but the most important are the number of hidden layers, number of neurons per hidden layer and activation function. These three lay the foundation for how complex and efficient a network should be, with effective tuning paying significant dividends in the model's overall performance. It is vital to keep a balance between number of neurons and number of layers, as overfitting can occur easily if not modified with moderation in relation to these parameters. This could make the model sensitive to outliers. Additional parameters used in the iterations include number of iterations which increases the model's chances of finding a solution; L1 & L2, which acts as a penalty to prevent overfitting, and number of tries, which allows the model to escape local max/min to reach global optimum.

Another trade-off is the in relation to the number of hidden layers and the compute time required for training. For instance, our dataset originally contained 17 attributes corresponding to 17 input neurons. Including all 17 attributes in the neural network may help the model reach optimal performance, however, in cases where the business problem dictates otherwise, it is important to omit where possible. In this case, as highlighted in section 5.3, Customer_ID was removed due to its lack of relevance.

To determine the best relative network, 13 iterations were conducted, as shown in appendix section 13.1.1. Each iteration was carried out with the same random seed of 12345 for selecting data observations to train on, preserving homogeneity. Although this is the case, it is important to note that re-initialised nodes with the same configuration may yield different results due to different random weight initialisations. The best model is shown below in Table 11.

NN Iteration Parameters												
Iteration No.	**No. of Hidden Layers**	**Neurons per Hidden Layer**	**Hidden Layer Activation Function**	**Number of Tries**	**Max No. of Iterations**	**L1**	**L2**	**Accuracy**	**KS**	**F1**	**ASE**	**AUC**
#1 NN-10	2	100	ReLu	1	600	0	0.01	0.8374	0.6805	0.8290	0.1177	0.9097

Table 11: NN – Best Performing Model

After completing all iterations, NN-10 was found to be the best performing model, with an accuracy of 83.74% and a KS score of 0.6805. Supplementary metrics such as F1 are also the highest, whilst ASE is the lowest. This is indicative of good relative performance, with less error. During the training and parameter tuning process, a few key insights

were discovered. Firstly, ReLu functions tend to outperform Tanh in most instances. This observation was first noticed in iterations 3 and 4, as shown in appendix section 13.1.1, with NN-4 performing drastically better than NN-3. Based on the same premise, NN-10 and 11 depicted a 3% difference in KS scores, with NN-10 outperforming iteration 11. Another observation was noticed in the Tanh activation function, where it was noticed to be less sensitive to changes in L1 and L2 penalties.

Increasing the L1 penalty in NN-9 caused the model to break, halting any predictions, which may be due to reaching a local maxima or minima. Iteration 12, 13 and 14 were experimental, and were used to try and push the model to its limits. Trying to increase the neurons per hidden layer above 100, broke these iterations, regardless of changes in other parameters. Following this, it was decided that 100 neurons per hidden layer was the optimal solution. The cause of this may be due to SAS compute limitations or due to an incompatible combination of parameters. A similar notion was exhibited by NN-14, where the best performing model's (NN-10) parameters were copied, with a change in max iterations to 700. This also caused the model to also fail, perhaps due to compute limitations. Overall, aside from broken iterations, most models achieved marginal differences in AUC values, with all iterations being within 1% of each other.

8.2. Logistic Regression

Unlike linear regression, logistic regression is a popular method for solving classification problems and resultantly was chosen as a technique for this dataset. However, given this difference, it is generally very robust as it can be used with both continuous and discrete data within the same model. As our dataset has a mix of both of these attribute types, it is compatible with generating predictions.

Logistic regression works by determining the relative significance of included effects or attributes, at each stage of attribute modification, based on various selection criteria. Using this premise, there are three most commonly used selection methods for outlining the training procedure relating to the inclusion, or deletion of attributes during regression. These are as follows:

- Forward Logistic Regression (FLR)
    - FLR is a regression methodology that starts with an empty model without any independent variables (predictors). With each iteration, FLR evaluates each predictor for its statistical significance and incrementally adds the most significant predictor to the model in each step. This iterative approach continues until no remaining predictors meet the inclusion criteria, resulting in a model that only includes relevant predictors, that show the greatest variation in the dependent variable, 'y'. It is important to note that once a variable is added for training due to its significance and satisfaction of inclusion criteria, it cannot be removed. The FLR process stops once no additional predictors meet inclusion criteria OR if selection stopping criteria is met.

- Backward Logistic Regression (BLR)
    - BLR works inversely to FLR, starting with a full model, meaning that all independent variables are included in regression from the first step. Each predictor is scrutinised iteratively, by calculating their relative significance and ability to demonstrate variation in the target variable – 'y'. This calculation will determine if they are removed or kept, based on internal rankings conducted by removing one variable at a time, and checking the drop in variance or significance. It is important to note that variables are checked one at a time during each deletion stage, before being added back to the model and continuing onto the next variable. At the end of the stage, once all variables that are present in the model at that time have been examined; the one with the least shift in variance or significance is removed. This is because it adds the least value in predicting if a potential client would sign up for a term deposit. Removed predictors cannot be added back to the model, with regression halting if selection stopping criteria is met.

- Stepwise Logistic Regression (SLR)
    - Unlike FLR and BLR, Stepwise Logistic Regression can freely add or remove predictors at each stage of regression. This means that all variables are evaluated at each step, with each variable being scrutinised for its ability to significantly impact the variance in the dependent variable – 'y'. If one variable is determined to be less significant in adding to the explanatory power of the regression model, it may be removed in early iterations. However, the same variable can also be added later if

its new effect on the dependent variable is greater due to the correlations between current variables in the model. For example, if two variables such as 'pdays' and 'previous' exist in the second step of SLR, 'pdays' may be omitted from the following regression step, as they are both correlated (highest bivariate correlation of 41% as shown in Figure 48). However, if 'previous' is removed in later regression steps, 'pdays' may be re-added to potentially demonstrate an increase in significance in the model's predictive power. Given SLR's ability to critique variables together, without being restricted in additive or subtractive functions; it tends to be more robust and is also the default selection method for the logistic regression node in SAS Viya.

To systematically test the three main types of logistic regression, a multitude of iterations will be executed, with slight modifications to some of the parameters tabulated below. These iterations are conducted with the intent to determine the best relative accuracy and KS scores for the regression classifier. As mentioned in section 7.1.4, it is also important to note that the random seed of 12345 will be kept constant across iterations to ensure reproducibility and homogeneity. Full descriptions of these parameters can be found in Appendix section 13.2.1.

Logistic Regression Parameter	Description
Effect Selection Criterion *Default: SBC (BIC)*	AIC, AICC, SBC (BIC), Significance level
Selection Process Stopping Criterion *Default: SBC (BIC)*	AIC, AICC, Validation Set (Average Squared Error), SBC (BIC), Significance level (default: 0.05), Validation ASE (Average Squared Error)
Max No. of Effects *Default: 0 (no max limit)*	The maximum number of predictor variables (effects) that will be allowed in the model at a given step.
Max No. of Steps *Default: 0 (option is ignored)*	Specifies the max no. of steps to complete during regression.
Optimisation Technique	Options include: None, Conjugate-gradient, Double-dogleg, Dual quasi-Newton, Nelder-Mead simplex, Newton-Raphson, Newton-Raphson w/ ridging, Trust-region.
Max No. of Iterations *Default: blank/NA*	By increasing this number, more optimisation refinement can occur for the selected optimisation technique, but at the cost of computational load.
Absolute function convergence	A small value that determines the forced stopping point irrespective of training.

Table 12: LR – Parameter Tuning

8.2.1. Forward Logistic Regression (FLR)

Tables 13 and 14 below contain the best iteration for FLR (according to KS score), with the rest found in Appendix section 13.2.2 for reference.

FLR Iteration Parameters							
Iteration No.	**Effect Selection Criterion**	**Selection Process Stopping Criterion**	**Max No. of Effects**	**Max No. of Steps**	**Optimisation Technique**	**Max No. of Iterations**	**Absolute function convergence**
FLR-13	SBC(BIC)	Validation ASE	17	30	Conjugate-gradient	130	0.0013

Table 13: FLR – Best performing model configuration

FLR Iteration Results (Test Set)					
#1 FLR-13	0.8119	0.6635	0.8124	0.1283	0.8959

Table 14: FLR – Best performing model – performance metrics

After model training, iteration 13 stood out as best performing model, based on the balance between highest KS score, accuracy, average squared error, and area under the curve. One interesting finding is that using SBC (BIC) for

both effect selection criteria and stopping criteria produced the best KS score and accuracy, with the lowest average square error. Trust-region optimization also performs competitively with Newton-Raphson (with ridging), within certain configurations. Conjugate-gradient has been used less frequently, but it still provides competitive results in certain configurations. Holistically, changing multiple parameters across iterations does not dramatically change model performance, with the lowest KS score being 65.03%, observed in iterations 4, 8, 14 and 18. Interestingly, these four iterations utilised the same effect selection criteria of 'significance level', perhaps indicating that this metric is more influential on the models' ability to further separate classes in their distributions, even with different optimisation techniques and iteration values.

The champion model (iteration 13) outperforms most iterations, with a KS score of 66.35%, indicating better separation than the lowest iterations described above; albeit minimal. While the accuracy for iteration 13 (81.19%) is marginally surpassed by iteration 3 (82.14%), when using analysing each performance metric carefully, it is evident that iteration 13 has a slightly higher AUC value, with a positive difference of 0.2%, and a lower ASE of 0.1283, compared to 0.1298. These disparities give iteration 13 the slight advantage in overall performance, enabling it to better predict if a client signs up for a term deposit, as validated by its marginally higher KS and AUC values.

The 20th iteration has the greatest number of effects, steps, and iterations of 24, 42 and 200, respectively. This may have reduced overfitting by revealing more complex relationships, with a trade-off for model training efficiency. Overall, F1 scores remained relatively consistent, with values ranging from 0.8124 to 0.8181. A pivotal metric to gauge a model's discriminative prowess, the AUC ranged from 0.8919 to 0.8959, suggesting good prediction certainty and coverage of yes/no targets across iterations.

To further critique the best performing FLR model (iteration 13), the same configuration was also run prior to tree based binning to determine if there were any changes in results. Iteration 23 was placed before binning, but used the same configuration as iteration 13, and performed slightly worse, with a KS score of 64.46%; nearly 2% lower than iteration 13. Therefore, it may be that SBC (BIC) is penalising FLR-23 more than FLR-13 due to its additional complexity without binned 'duration' and 'balance' attributes.

8.2.2. Backward Logistic Regression (BLR)

Tables 15 and 16 below contain the best iteration for BLR (according to KS score), with the rest found in Appendix section 13.2.3 for reference, containing all 12 iterations. One important change in BLR from FLR, is the shift from max. no. of effects to min. no. of effects. This is due to the deletion process involved in BLR, as it begins with a full model.

BLR Iteration Parameters							
Iteration No.	**Effect Selection Criterion**	**Selection Process Stopping Criterion**	**Min No. of Effects**	**Max No. of Steps**	**Optimisation Technique**	**Max No. of Iterations**	**Absolute function convergence**
BLR-5	SBC (BIC)	Validation ASE	0	0	Trust-region	95	NA

Table 15: BLR – Best performing model configuration

BLR Iteration Results (Test Set)					
Iteration No.	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under the Curve**
#1 BLR-5	0.8166	0.6578	0.8173	0.1279	0.8969

Table 16: BLR – Best performing model – performance metrics

After completing 12 iterations, it is clear that BLR-5, as shown in Tables 15 & 16 above, is the best performing model. BLR-5 boasted a KS score of 65.78%, with the lowest ASE of 0.1279, and an AUC of 89.69%. These metrics suggest that the model has relatively good separation of yes and no classes in the target variable, 'y', and is also able to create a relatively high probabilistic differentiation between them, given the highest AUC score. Interestingly, BLR-5's performance metrics are exactly matched by BLR-10's; potentially indicating that changes in stopping criteria from validation ASE to AIC, reducing the iterations to 50 (default) and including a forced convergence number, did not significantly impact the model they converged on. Analysing the SBC values for both these iterations also unveils

interesting findings. As shown in Figures 60 and 61, BLR-5 has an SBC value of 1185, whilst BLR-10 is 1192. This is logically sound, as BLR-10 was forced to use at least 5 minimum attributes for model convergence, with lower trust-region iterations, and predictability, it failed to outperform BLR-5. This reinforces the importance of omitting hardcoded effect and convergence values, as the chances of generating a lower complexity model are relatively low.

Statistic	Testing
M2LL	886.0685
AIC	972.0685
AICC	975.8003
SBC	1,185.5263
ASE	0.1279

Figure 60: BLR-5 SBC Value

Statistic	Testing
M2LL	886.0341
AIC	974.0341
AICC	977.9433
SBC	1,192.4560
ASE	0.1279

Figure 61: BLR-10 SBC Value

Another trend was discovered, during the comparison between the default models BLR-1 and BLR-11, with BLR-11 being trained prior to binning. Comparatively, BLR-11 reported a marginal loss across all performance metrics, reinforcing the notion initially mentioned in FLR, where SBC and logistic regression may be penalising attributes with higher unique values (levels), since they are not binned. BLR-12 was used to corroborate this, by using the best model (BLR-5), but placing it in the pipeline, prior to binning 'balance' and 'duration' attributes. It was again found that, BLR-12 has lower performance metrics than BLR-5 across the board, confirming that logistic regression is heavily dependent on the dimensionality of input data, with penalties increasing in relation to the number of unique levels within attributes.

8.2.3. Stepwise Logistic Regression (SLR)

Tables 17 and 18 below contain the best iteration for SLR (according to KS score), with the rest found in Appendix section 13.2.4 for reference, containing all 12 iterations. As stepwise is fundamentally different in its evaluation procedure, out of FLR and BLR, backward logistic regression was chosen as a benchmark for SLR to measure against. To ensure homogeneity, the same iteration configurations will be employed for SLR to gauge if any differences occur between both regression methods. One important change in SLR from BLR is the shift from min. no. of effects to max. no. of effects. This is due to the stepwise addition/removal process involved in SLR, as it begins with an empty model.

SLR Iteration Parameters							
Iteration No.	**Effect Selection Criterion**	**Selection Process Stopping Criterion**	**Max No. of Effects**	**Max No. of Steps**	**Optimisation Technique**	**Max No. of Iterations**	**Absolute function convergence**
SLR-1 (Default)	SBC (BIC)	SBC (BIC)	0	0	Newton-Raphson with ridging	NA (50)	NA

Table 17: SLR – Best performing model configuration

SLR Iteration Results (Test Set)					
Iteration No.	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under the Curve**
#1 SLR-1 (Default)	0.8157	0.6635	0.8176	0.1299	0.8930

Table 18: SLR – Best performing model – performance metrics

After 12 iterations, it is evident that SLR-1,5,7 and 10 matched perfectly, with KS and Accuracy scores of 66.35% and 81.57%, respectively. Moving forward, SLR-1 will be referred to as the best model, and when compared to BLR-5, SLR-1 narrowly underperforms in all metrics except KS score, with a 0.57% increase. Comparing this to FLR's best iteration – FLR-13; the results become more convoluted, with SLR-1 scarcely outperforming it in accuracy, F1 and AUC scores, but portraying a slightly higher ASE of 0.1299, compared to FLR-13's 0.1283. Evidently, all best performing logistic regression models are similarly ranked, and do not exhibit drastic score differences between them. Generally, SLR was originally predicted to be the best performing model due to its stepwise attribute

evaluation procedure, exploiting the ability to re-add omitted attributes at a later stage. However, given the dataset's overall low correlation variance, and lack of distributed importance as shown in Figures 48 and 49, stepwise logistic regression may not have been able to discover strong hidden insights, that created significant variance in the target variable within SLR-1. Due to this, the inherent nature of SLR to add and remove attributes based on their importance, correlation, and causal variance in the target attribute, may not have been drastically beneficial, even if more compute power was needed.

Another interesting observation were the matching results for SLR-11 and 12, that both occurred prior to binning. When comparing SLR-1 and 12's t-value distributions as shown in Figures 62 and 63 below, it is discernible that duration was by far the most significant attribute that caused the most variance in the target attribute. Even though SLR-12 had this characteristic in its model, it did not outperform SLR-1, perhaps due to it slightly overfitting on specific values, instead of generalising via the binned version used in SLR-1. Comparatively, SLR-1 found that 'poutcome_success', exhibited the greatest change in the target variable, 'y', with the two Figures below generally matching each other after this category. This is significant, as it is clear that a successful outcome from the client's previous marketing campaign is very likely to influence their decision to sign up for a new term deposit. Similarly, clients that were contacted via cellular services also seem to have a significant influence on the target variable, which corroborates the multivariate analysis conducted in section 4.2.5. This section suggested that cellular communication may have been preferred by the bank due to a higher likelihood of answering the phone.

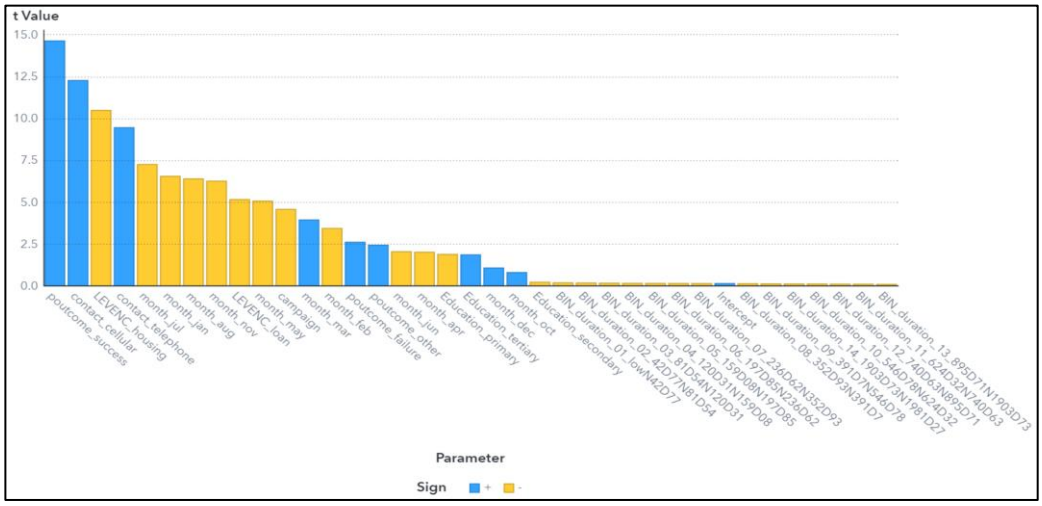


Figure 62: SLR-1 t-value distribution

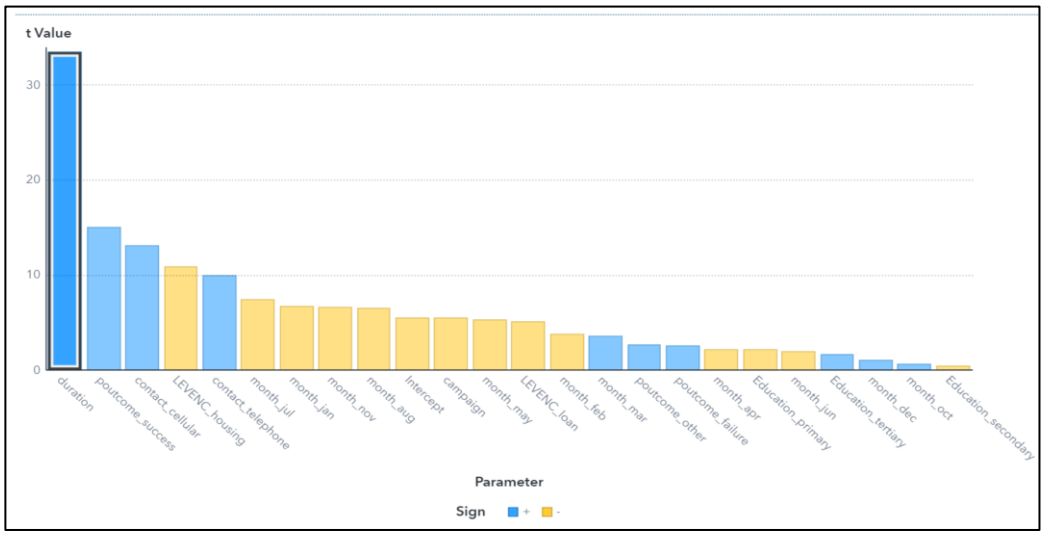


Figure 63: SLR-12 t-value distribution

Lastly, SLR-2,4,6 and 8 exhibited matching KS, F1 and accuracy scores, which reinforced the prediction mentioned at the end of section 7.1.4, where KS and accuracy scores generally move together, given that the 'BANK_DIRECT_MARKETING' dataset is perfectly balanced in its target attribute distribution, with 50% yes and 50% no values.

8.3. Decision Tree

A decision tree is a flowchart-like structure where an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents an outcome of the decision. The node at the top of the decision tree is called a root node which is based on the feature that results in the highest information gain. A decision tree learns to partition the data based on the attribute values, in a recursive manner.

Decision trees are chosen for this project due to its fitting nature for the BANK_DIRECT_MARKETING dataset. The dataset consists of both numerical and categorical variables and decision trees can handle both type of data seamlessly without the need for explicit transformation. For business decisions, it is important to explain the reasoning behind a prediction, favouring decision trees, as they offer an intuitive visual representation for various stakeholders. Given the nature of the dataset, a decision tree can mirror the real-life decision-making process of determining whether a customer is likely to subscribe to a term deposit or not.

To iteratively test the decision tree classifier, the following parameters will be tweaked to determine the best performing model in terms of KS and accuracy scores. A random seed of 12345 was used across all iterations to ensure homogenous testing. A more detailed description of these parameters can be found in section 13.3.1 in the appendix.

Decision Tree Parameter	Options
Class Target Criterion *Default: Info Gain Ratio*	Info Gain Ratio, CHAID, Chi-square, Entropy, Gini
Interval Target Criterion *Default: Variance*	CHAID, F-test, Variance
Bonferroni *Default: Deselected*	Yes/No
Max. No. of Branches *Default: 2*	Value Range from 2-10
Max. Depth *Default: 10*	Value Range from 1-150
Min. Leaf Size *Default: 5*	Value Range from 1-2147483647

Table 19: DT – Parameter Selections

Tables 20 and 21 below contain the best iteration for the decision tree classifier (according to KS score), with the rest found in Appendix section 13.3.2 for reference, containing all 21 iterations.

DT Iteration Parameters						
Iteration No.	**Class Target Criterion**	**Interval Target Criterion**	**Bonferroni**	**Max. No of Branches**	**Max. Depth**	**Min. Leaf Size**
DT-21 (Best model DT-15, prior to binning)	Entropy	CHAID	Disabled	2	10	5

Table 20: DT – Best performing model configuration

DT Iteration Results (Test Set)					
Iteration No.	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under the Curve**
#1 DT-21 (Best model DT-15, prior to binning)	0.8327	0.6654	0.8395	0.1301	0.8819

Table 21: DT – Best performing model – performance metrics

Throughout the 21 iterations, a mix of various class target criteria were used such as IGR (Information Gain Ratio), Gini, and Entropy. Following this, models were systematically tested with different interval target criteria such as Variance, F Test and CHAID. Upon completion of the preliminary iterations, three best performing models were

selected with the highest accuracies, which were then modified based on their number of branches, depth, and leaf size. After running all iterations with these changes, iteration 15 was found to be the best performing model after tree-based binning. It was configured with balanced parameters, using a combination of entropy for class targets and CHAID for interval targets. A max. depth of 10 and a min. leaf size of 5 was employed, which may have resulted in a more balanced tree that neither overfits nor underfits the training data, leading to better test set generalisation. By using CHAID, the model bases its decision on statistical significance which potentially lead to nodes that have actual practical significance rather than just mathematical optimisation. Entropy also ensures that the decision at each node is made to maximise the information gain, which leads to a tree that can make decisions based on significant differences in the data. This combination with the appropriate depth and leaf size settings potentially helped with pruning to ensure overfitting does not occur.

To benchmark DT-15, DT-21 was trained using the same configuration, but prior to tree-based binning. This resulted in an optimal model that exhibited better KS and accuracy characteristics than DT-15, with a KS score of 66.54% and an accuracy of 83.27%. By omitting the binning process, it is likely that DT-21 captured fine-grained details and variations in the data, which allowed it to be more sensitive towards subtle differences between data points, providing an accurate representation of the data. While binning simplifies input data, raw data generates a more complex model, which may lead to better performance as important nuances in the data can be captured. The boost in accuracy within DT-21, may also relate to the fact that tree-based binning does not explicitly consider attributes in the dataset, and how their relationships in future node splits could impact the significance of binned data. This may be another reason for the relatively poor performance of DT-15, in comparison to DT-21.

8.4. Gradient Boosted Tree

Gradient Boosted Tree (GBT) is an advanced ensemble-based machine learning algorithm that stands out in classification and regression tasks. This technique's essence lies in its iterative approach, where each model in the sequence works to correct the shortcomings of its predecessor, using the principle of gradient descent optimization (Natekin & Knoll, 2013). Central to Gradient Boosting are decision trees, often shallow in nature. Their simplicity ensures swift computation and provides a safeguard against overfitting. These trees systematically evaluate data inputs based on decision nodes, making them effective in capturing broad patterns while remaining computationally efficient. Considering the BANK_DIRECT_MARKETING dataset, its composition of both numerical and categorical variables, illustrates the adaptability of Gradient Boosting. For instance, numerical features like 'Age', 'balance', and 'duration' offer continuous data, potentially revealing trends related to a customer's financial stability or engagement with the bank. Meanwhile, categorical variables such as 'Education', 'JOB', or 'contact' can be seamlessly handled, enabling the model to differentiate and weigh the influence of diverse customer profiles and their communication preferences.

One of GBT's attributes is its flexibility in managing different data types. Whether it's 'marital' status or binary variables like 'loan' and 'housing', GBT's capability to discern and classify based on these diverse inputs is highly preferred, which contributed to its inclusion in predicting term deposit signups. However, the performance of GBT is heavily contingent upon its hyperparameter configuration. The learning rate determines how aggressively errors are adjusted; the number of trees indicate the model's complexity, whilst their depth, reflects decision node granularity. Striking the right balance among these parameters, especially with rich datasets such as BANK_DIRECT_MARKETING, is essential to harness the true power of Gradient Boosting.

To ensure each parameter is tested appropriately a series of iterations will use the following tabulated parameters to determine the best relative accuracy and KS scores for the GBT classifier. Detailed explanations of these parameters can be found in Appendix section 13.4.1. To be consistent with other classifiers, a random seed value of 12345 will be used.

Gradient Boosted Parameter	Options
No. of Trees *Default: 100*	Value Range from 1-10000
Learning Rate *Default: 0.1*	Value Range from 0.1-1
Subsample Rate *Default: 0.5*	Value Range from 0.1-1
L1 regularisation *Default: 0*	No limit

L2 regularisation *Default: 1*	No limit

Table 22: GBT – Parameter selection

Table 23 below contains the best iteration for the gradient boosted classifier, with the rest found in Appendix section 13.4.2 for reference, containing all 21 iterations. It is important to note that all GB iterations were conducted with a max branch value of 2, and a max depth of 15.

GB Iteration Parameters										
Iteration No.	**No. of Trees**	**Learning rate**	**Subsample rate**	**L1 regularisation**	**L2 regularisation**	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under Curve**
#1 GB-21 (Best model - prior to binning)	150	0.085	0.9	0.25	1.7	0.8478	0.6957	0.8546	0.1174	0.9087

Table 23: GB – Best performing model

Gradient boosting, renowned for its adaptability, employs an ensemble technique that progressively refines models based on previous trees' errors. Extensive testing involved varying parameters such as the number of trees, learning rate, subsample rate, and L1 and L2 regularisation, to elucidate their influence on diverse metrics. With the KS score as our foremost metric, complemented by accuracy and others, our evaluation aims to provide holistic insights into the model's performance. Starting with the KS score, which evaluates a model's ability to separate classes; the data unveils interesting observations. The 21st iteration attains the highest KS score of 0.6957, edging out its closest competitors: iterations 10, 14, and 16, which all gravitate around the 0.69 range. This metric's change across iterations highlights the sensitivity of gradient boosting to parameter changes, albeit minimal. Examining accuracy, our secondary metric, the narrative is further corroborated. Iteration 21, with an accuracy of 0.8478, reiterates its dominance, exemplifying consistency across primary decision metrics. This congruence between KS score and accuracy in certain iterations, like the 21st, elucidates the alignment of general model accuracy with its specific discriminatory prowess, as originally mentioned in section 7.1.4 – model training and analysis.

During training, iteration 1 was set with default parameters to act as a control for default iterations with tree-based binning. Iteration 20 utilised this same config, but was trained without tree-binning. Whilst the 20th iteration mirrored parameters from the 1st iteration, it revealed a marginal decline in both the KS score and accuracy, hinting at tree-binning's contributory role. This narrative is counteracted by iteration 21. By excelling in both KS score and accuracy without tree-binning, it advocates for the assertion that optimal results are attainable even without this preprocessing step, contingent on the synergy of other parameters. This shows there is no causal probability that tree-based binning affects the output of the tested gradient boosted models. Examining ASE, it is evident that iteration 8, with the lowest average squared error of 0.1141, underscores its capability to produce predictions closely aligned with actual values, minimising deviation. Additionally, the F1 score, which combines precision and recall, finds its pinnacle in iteration 21. This reaffirms GB-21's performance, suggesting its potential efficacy even in scenarios where the costs of false positives and false negatives are substantive.

Analysing different configurations, we observe diverse combinations. For instance, iteration 6, with 85 trees, a learning rate of 0.035, and a high subsample rate of 0.75, achieved a respectable KS score of 0.6881 and an accuracy of 0.8440. This portrays the ensemble technique's resilience to varied configurations, reinforcing its likely robustness. Overall, iteration 21 depicted the best performance, while others like 20 and 8 provide insights into the complexities and nuances of model tuning, as shown Appendix section 13.4.2. The oscillations in performance metrics across iterations, emphasise the interplay between parameter tuning, preprocessing choices, and intrinsic algorithmic characteristics in shaping the model's outcomes; resulting in marginal differences across iteration metrics.

8.5. Random Forest

A random forest classifier is built upon the more standard decision tree model, and thus employs the same recursive node structure of a rule-based tree. Each decision tree within the random forest contains a root node containing the first rule to segregate the input data into 2 binary categories. This tree-split algorithm employed by SAS Viya

recursively creates more rule nodes to further split the data until only leaf nodes remain. These nodes have no further splits available due to a lack of available rules or data. As the BANK_DIRECT_MARKETING dataset is a binary classification problem, a single decision tree will output either 1 or 0 depending on the decision tree's prediction on if the potential client will sign up for a term deposit.

Whilst decision trees are simple in nature and are great for datasets with minimal variance, they struggle when their input data is significantly different from the larger dataset. This also makes them prone to overfitting, especially when the tree depth (layers of nodes) is too high. To combat this, using multiple trees in an ensemble can increase prediction accuracy by using randomised subsets of the training data for each tree (bootstrap aggregation), while also using randomised rule splits for each respective tree. This random nature reduces variance or biases introduced by training on the same subset of data, and provides a majority voting opportunity for all trees in the ensemble to select the mode (most occurring) predicted label. In this case, the majority of trees will output either 'yes' for a potential customer signing up for a term deposit, and 'no' if they do not.

Given these aggregate benefits, the random forest classifier was chosen to predict the target variable 'y' for the BANK_DIRECT_MARKETING dataset, in an attempt to correct the downsides of single decision tree models. It will be interesting to compare these results with those from the decision tree classifier, to determine any significant insights between their accuracies and/or KS scores.

To thoroughly test the random forest classifier, 15 iterations will be conducted, with slight changes to the parameters tabulated below. These iterations are conducted with the intent to determine the best relative accuracy and KS scores for the random forest classifier. As mentioned in section 7.1.4, it is important to note that the random seed of 12345 will be kept constant across iterations to ensure reproducibility and homogeneity. Detailed descriptions of these parameters can be found in Appendix section 13.5.1.

Random Forest Parameter	**Options**
No. of Trees *Default: 100*	Number of trees in the random forest ensemble
Class Target Voting Method *Default: Probability*	Probability, Majority
Class Target Criterion *Default: Info Gain Ratio*	CHAID, Chi-square, Entropy, Gini, Info Gain Ratio.
Max. No. of Branches *Default: 2*	Ranges from 2-5.
Max. Depth *Default: 20*	Ranges from 1-50.
Leaf Size Specification *Default: Count*	Count, Proportion
Min. Leaf Size (used when COUNT is selected for leaf size specification) *Default: 5*	Smallest No. of observation present in a new branch.
Min. Leaf Proportion (used when PROPORTION is selected for leaf size specification) *Default: 0.00005*	Smallest fractional proportion of observations with an available target value.
Interval Bin No. *Default: 50*	Number of bins
Binning Method *Default: Quantile*	Bucket, Quantile
In-bag sample proportion *Default: 0.6*	Training subset proportion

Table 24: RF – Parameter selection

Tables 25 and 26 below contain the best iteration for the random forest classifier, with the rest found in Appendix section 13.5.2 for reference, containing all 21 iterations.

RF Iteration Parameters											
Iteration No.	No. of Trees	Class Target Voting Method	Class Target Criterion	Max. No. of Branches	Max. Depth	Leaf Size Spec	Min. Leaf Size	Min. Leaf Proportion	Interval Bin No.	Bin Method	In-bag proportion
RF-17	900	Majority	Info Gain	5	35	Count	5	NA	50	Quantile	0.7

Table 25: RF – Best model configuration

Random Forest Iteration Results (Test Set)					
Iteration No.	Accuracy	KS score	F1 score	Average Squared Error	Area Under the Curve
#1 RF-17	0.8507	0.7013	0.8577	0.1152	0.9096

Table 26: RF – Best model configuration – performance metrics

Upon completion of 21 iterations, the highest accuracy and KS scores achieved were 85.07% and 70.13%, respectively. The top three iterations according to KS score and accuracy, are marked in green, orange, and red, in accordance with the 1st, 2nd and 3rd iterations, respectively. It is interesting to note that these three iterations utilised the same class target criterion of information gain ratio. This may be indicative of the criterion's strengths in handling multi-valued attributes such as 'duration' and 'poutcome', which were found to be the 1st and 2nd most important attributes as shown in Figure 49, with duration also having a plethora of unique values totalling to greater than 254. Examining iterations 1 and 13 from appendix section 13.5.2, it is also evident that with a default configuration random forest node, there is a difference shown when running the ensemble classifier after and prior to tree-based binning of duration and balance, respectively. Iteration 1 exhibits a slightly lower KS score of 68.43%, whilst the removal of tree-based binning prior to training for the same configuration (iteration 13), resulted in a 0.57% increase. Albeit minor, this positive shift could stipulate that the lack of prior interval binning assists in providing more nuanced data, and therefore results in a marginally better separation between 'yes' and 'no' target classes, given the KS score of 69% in iteration 13.

Contrastingly, the inverse is discovered during iterations 17 and 20, as tabulated in appendix section 13.5.2. With iteration 17 being the highest performing model with a KS score of 70.13%, its ability to separate the 'yes' and 'no' target classes, is more than 1.5x greater than the worst performing model in iteration 4, with a KS score of 43.29%. To challenge this, iteration 20 was used to benchmark this score by conducting training prior to tree based binning, but with the same configuration. Interestingly, iteration 20's KS score was marginally lower than iteration 17's by 0.38%. This is in stark contrast to the inverse phenomenon encountered with the default random forest node in iterations 1 and 13. This may be due to the 9x increase in tree count from 100 to 900 within iterations 17 and 20. The high tree count, paired with quantile binning may have been able to overcome challenges associated with a lack of nuanced data, and instead generated marginally more effective rule splits, albeit at the cost of training efficiency. This notion was further reinforced in iteration 21, where the same configuration as the best model (iteration 17) was used, with a change to bucket binning. This resulted in a 1.51% decrease in KS score and a 0.95% drop in accuracy when compared to the highest scoring model trained during iteration 17. This observed disparity could be due to the inherent nature of bucket binning, which groups data according to where they fall into a min-max range, instead of equally distributing data points within categories, to create the potential for a better rule split calculation (due to less data clustering in a single bin).

It was also observed that the CHAID criterion was used for iterations 4 and 12, which were the worst performing models. This finding may be due to CHAID's requirements of needing pure categorical data, forcing it to group interval attributes such as balance and age into unordered bins (Singh, 2021; Kumari, 2022; Bhalla, n.d.; Roberts, 2023). This may have stripped intrinsic information from these independent attributes, equating to the lowest accuracy of 71.64% observed in iteration 4, as shown in appendix section 13.5.2. This may be further reinforced by the ROC graph for iteration 12, as shown in Figure 64 below, where RF-12 is almost perfectly achieving a 100% true positive prediction rate on the training set, depicted by the close proximity to the y-axis towards the upper left corner. This is an indication of severe overfitting and is fortified by its poor test set ROC curve shown in Figure 65 below. This notion is corroborated by its inferior AUC value of 84.81%; only made worse by iteration 4. As a final note, it was also discerned that there was a general trend in high KS values being mirrored by high accuracy values, validating the initial prediction made at the end of section 7.1.4 – model training & analysis.

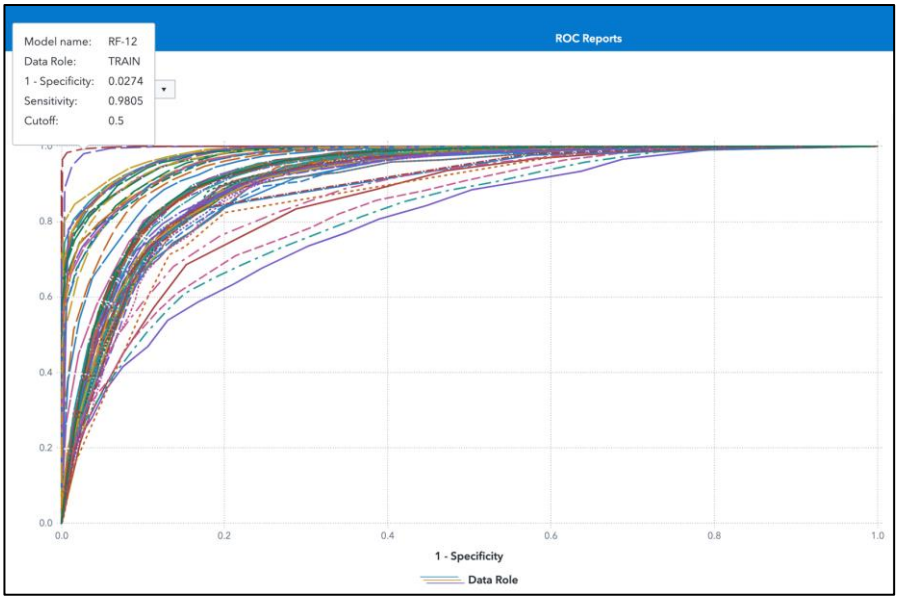


Figure 64: RF-12 ROC Curve – Train set

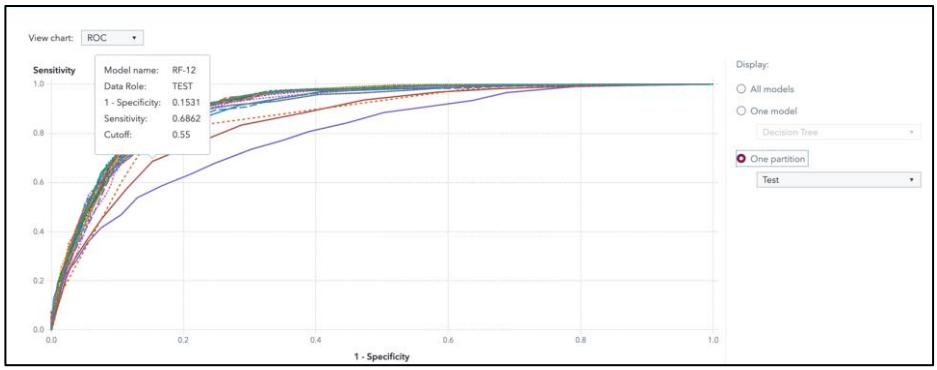


Figure 65: RF-12 ROC Curve – Test set

8.6. Support Vector Machine

The SVM was developed to classify the outputs with no intention of understanding the complex relationship between them. In a more advanced version of SAS, this model also allows the utilisation of ranks and estimates of probability. However, the model's primary focus is binary classification due to the nature of our dataset, where predictions on customer term deposit signups are paramount.

SVM can work with a very simple datasets, when using the linear kernel function. However, more complex datasets with more attributes require polynomial kernels to comprehend and predict patterns. SVM works using this kernel by mapping input data points to a high dimensional feature space, where data points can be categorised without the need for linear separation. Once completed, SVM transforms the data so that a hyperplane can be used to categorise the transformed data more easily. Since the BANK_DIRECT_MARKETING dataset has 16 usable attributes, it is predicted that the most appropriate model is a hyperplane classifier with a Polynomial Kernel.

To thoroughly test SVM models, it is important to consider the following key parameters:

- Penalty: higher values penalise misclassification greater than lower values
- Kernel: set of functions used by SVM to transform the data
- Tolerance: determines the minimum number at which the iteration halts
- Max iterations: max iterations per classification attempt

Detailed descriptions of these parameters can be found in appendix section 13.6.1.

Using these parameters, 22 iterations were conducted, with slight changes, as shown in appendix section 13.6.2. These iterations were undertaken with the intent to determine the best relative accuracy and KS scores for the SVM classifier. The table below shows the configuration for the best performing model.

SVM Iteration Parameters									
Iteration No.	**Penalty**	**Kernel**	**Tolerance**	**Max iteration**	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under Curve**
#1 SVM-22	0.1	Poly (2)	0.1	5	0.8346	0.6692	0.8299	0.1580	0.9037

Table 27: SVM – Best model configuration – performance metrics

Upon completion of 24 iterations, SVM-22 was chosen as the best model, with a KS score of 0.6692, and accuracy of 83.46%. Within the first 15 iterations, penalties of 0.1 were used to find optimal hyperparameters to generate the highest KS score. After these 15 iterations, SVM-11 exhibited the highest KS score of 66.92%. Other evaluation metric such as F1 score, and AUC was also relatively competitive. Interestingly, iterations 8 and 14 as shown appendix section 13.6.2, matched perfectly in their results, suggesting that common penalty and kernel values are more significant to the convergence of the model, instead of more iterations or tolerance. Similarly, iterations 10 and 13 also corroborate this finding, with the number of iterations not increasing prediction accuracy any further. This suggests that the SVM model has already found the optimal hyperplane, given the penalty, kernel and tolerance parameters.

From iteration 18 to 24, the parameters from iteration 11 were modified by changing them independently to observe any changes. Ultimately, three iterations during testing tied 1st with the highest KS score of 0.6692. These are SVM-11,16 and 22. This may be due to the use of the same kernel function (Poly – 2) for all three, again highlighting that even with their differences in iterations ranging from 5 to 25, kernel function is of more importance. Many iterations used kernel values of Poly – 2, paired with a tolerance of 0.1. Using this as a benchmark for the accuracy seen in the aforementioned iterations (SVM-11,16,22), it would be noteworthy to change the penalty and increase the maximum iterations to observe differences. Completing this change uncovered the decrease in KS score for iteration 24 as shown in appendix section 13.6.2, caused by a higher penalty, as originally predicted. Overall, the distribution of results is marginally different across all iterations without drastic spikes.

9. Comparative model assessment

Adhering to the data mining process described particularly in sections 7.1.4 and 7.1.5, systematic intra-model evaluations were undertaken in respective classifier sections above. Each classification technique was thoroughly examined, with ranking emphasis placed on KS score and accuracy. Additional relative performance measures such as average squared error, F1 score, and area under the curve were examined where relevant. By reducing the scope to examine each individual classifier separately within their iterations, it was apparent that the percentage range, or variation between each performance metric across iterations, was marginal at best. This may suggest that each model reached relative convergence, a potential local minima/maxima, or may have been hindered through unseen errors in the preprocessing pipeline described in section 5. Although the latter is unlikely given that all best classifier models surpassed our initial accuracy goal of at least 75-80%, it is important not to exclude the potential for any unforeseen erroneous preprocessing. The percentage range (highest % value – lowest % value) mapping of these performance metrics across all models is tabulated below.

Percentage ranges for intra-model performance metrics					
Classification technique	**KS score**	**Accuracy**	**F1 score**	**Average Squared Error**	**Area under the Curve**
Neural Network	68.05%	33.74%	82.90%	13.23%	40.97%
Forward Logistic Regression	3.97%	1.42%	1.89%	0.51%	0.52%
Backward Logistic Regression	2.64%	1.32%	1.68%	0.44%	0.80%
Stepwise Logistic Regression	3.97%	1.04%	1.35%	0.26%	0.39%
Decision Tree	4.16%	2.08%	2.53%	1.41%	2.28%
Gradient Boosted Tree	3.41%	2.17%	2.25%	2.16%	1.08%
Random Forest	26.84%	13.43%	13.60%	7.23%	11.36%
Support Vector Machine	7.75%	3.88%	3.68%	3.81%	4.73%

Table 28: Percentage range differences across best performing models

Examining the table, it is clear that the percentage range variance within logistic regression, decision tree and gradient boosted models are negligible, suggesting that they are relatively consistent. These values could relate to

the fact that these models have explicit penalty values, perhaps creating a modicum of regularisation across all of them.

Contrastingly, the neural network ranges are the most drastic in the table, and were due to compute limitations encountered during model training. This may have caused the lowest models to get trapped in a local minima or maxima. Random forest also exhibited relatively large variances, which are explained by RF-4, as shown in appendix section 13.5.2. RF-4 utilised the CHAID criterion, which when used, resulted in the worst RF models. It was discovered that this could be a resultant of CHAID's requirements of needing pure categorical data, forcing it to group interval attributes such as balance and age into unordered bins (Singh, 2021; Kumari, 2022; Bhalla, n.d.; Roberts, 2023). This may have stripped intrinsic information from these independent attributes, equating to the lowest accuracy of 71.64%, as observed in iteration 4. These discrepancies are exemplified in Table 28 above, with the largest range being described by neural network and random forest models, respectively.

Leveraging these intra-model performance metrics, comparisons between the highest performing classification techniques can be undertaken. To ascertain the best models within each classification technique, they were ranked in accordance with their KS scores, and their relation to accuracy. KS scores assess a model's capability to distinguish between 'yes' and 'no' targets, alluding to their prediction certainty and refers to the ability to avoid misclassification. This is a particularly good metric if targets are closely positioned for certain data points. This combination was intentionally chosen as the primary intra-model ranking criteria, due to their combined ability to detect severe imbalances in target class variables, as detailed in section 7.1.4. Severe imbalances can be observed with low accuracy scores and high KS values, which again reinforces their robustness as a combined metric for intra-model evaluation. However, as the target variable, 'y' is balanced evenly in the BANK_DIRECT_MARKETING dataset, this is not an issue. Nevertheless, accuracy was used to corroborate KS scores throughout multiple iterations. As stated in section 7.1.4, it was originally predicted that ranking KS scores from high to low, would also in effect, **generally** rank models by accuracy (high to low), as they are intrinsically related. This prediction proved to be true across all iterations, again accentuating the robustness of these two metrics, regardless of inverse or even targets distributions.

Upon completion of this iteration analysis, the best performing model from each classifier was selected and is tabulated below, sorted by descending KS score.

Performance metrics - Best performing models (test set)							
Classification Technique	**KS score**	**Accuracy**	**F1 score**	**Average Squared Error**	**Area under the Curve**	**Gini coefficient**	**SBC**
#1 - Random Forest (RF-17)	70.13%	85.07%	85.77%	11.52%	90.96%	0.8192	N/A
#2 - Gradient Boosted (GB-21)	69.57%	84.78%	85.46%	11.74%	90.87%	0.8140	N/A
#3 - Neural Network (NN-10)	68.05%	83.74%	82.90%	11.77%	90.97%	0.8192	N/A
Support Vector Machine (SVM-22)	66.92%	83.46%	82.99%	15.80%	90.37%	0.8074	N/A
Decision Tree (DT-21)	66.54%	83.27%	83.95%	13.01%	88.19%	0.7639	N/A
Forward Logistic Regression (FLR-13)	66.35%	81.19%	81.24%	12.83%	89.59%	0.7874	1135.0640
Stepwise Logistic Regression (SLR-1)	66.35%	81.57%	81.76%	12.99%	89.30%	0.7860	1151.9346
Backward Logistic Regression (BLR-5)	65.78%	81.66%	81.73%	12.79%	89.69%	0.7937	1185.5263

Table 29: Best performing models ranked by KS score

Derived from the table, it is clear that the random forest classifier is superior to all others, boasting a KS score of 70.13%, and an accuracy of 85.07%. Examining the results of GB-21 is particularly intriguing, as it was trained on the dataset prior to tree-based binning of duration and balance attributes. This injected more unique data points with relatively high variances. GB-21 was adapted from GB-16, as shown in appendix section 13.4.2. GB-16 had the same

configuration, but was trained after binning of these two attributes, and achieved a KS score of 69.19% and accuracy of 84.12%. Interestingly, using the same settings in GB-21 prior to tree based binning, an increase in KS and accuracy scores were observed. Albeit marginal, this may be due to the model's ability to make better decision splits with more data, assisting in tuning the classification error in each sequential tree that is generated. However, while there is a slight increase in performance metrics for GB-21, compared to GB-16, the inclusion of the two variables with relatively high kurtosis scores of 119.65 for balance and 7.36 for duration, may have counteracted and stunted the potential gain achieved by binning omission.

These values are indicative of a platykurtic distribution, exhibiting sharp peaks and heavy tails, as depicted in sections 3.2.7 and 3.2.13, for balance and duration, respectively. The nature of the distribution highlights the inclusion of extra outliers in the raw data for these two attributes, and may have hindered GB-21 from reaching higher scores, as error residuals may have been carried forward during each sequence, exemplifying errors over time. This may also be corroborated by GB-21's 150 tree value, being the 2nd highest value for this parameter, which could have caused this marginal discrepancy. Nevertheless, as the difference between RF-17 and GB-21 is miniscule, this may not be the only reason for lower metrics, and instead could be due to stochastic subsample fluctuations during initial tree training (Overview of Gradient Boosting, n.d.). Since these exact random samples are not inherently stored within the node upon upload, replication is unlikely, with exact subsample values not being reproducible regardless of seed values.

The best neural network classifier came in at 3rd place, with an accuracy score of 83.74% and a KS score of 68.05%. With these scores being negligibly close to the top two models, explicit justifications for its relative decline are difficult to describe with certainty. However, given the sheer number of hyperparameters available for modification in the neural network node, a multitude of iterations would be required to effectively test a plethora of combinations. Without the ability to leverage k-fold cross validation or autotuning loops, this was not feasible. Additionally, as the dataset only contains 10,578 records, which may have caused NN-10 to marginally overfit on the training data, as the model may not have had enough variance in input data.

Aside from these metrics, comparing Gini values also provides further insight into the top classifiers during inter-model analysis. RF-17 and GB-21 present logical Gini values, with the former having a value of 0.8192, and the latter having a value of 0.8140. This corroborates their individual AUC values, with GB-21 seeing a decrease in both, highlighting AUC's subsidiary role in determining Gini coefficients. This again showcases RF-17's ability to create a marginally better differentiation between the positive class ('yes') or negative class ('no') in relation to term deposit registration. Interestingly, NN-10's Gini value is the same as the best performing classifier, RF-17, yet has a lower accuracy score of 83.74%. This may be due to the fact that AUC, and by derivation Gini is computed by generally adding accuracies at all possible thresholds, whilst accuracy only performs its calculation on a threshold of 0.5 for binary classification. This indicates that NN-10 has the potential to select a better threshold cut-off value for separating classes (Herbsthofer, 2021). This narrative is also corroborated by Figure 66 below, where NN-10's test set accuracy chart peaks at a cut-off of 0.55, instead of 0.5. This simple change would increase the accuracy from 83.74% to 84.03%, with the potential for a higher increase following further iterative parameter tuning. Due to SAS Viya limitations, *Cutoff Nodes* ("Cutoff node," n.d.) are not included in standard node selection, so modifications to this threshold cannot be executed manually.

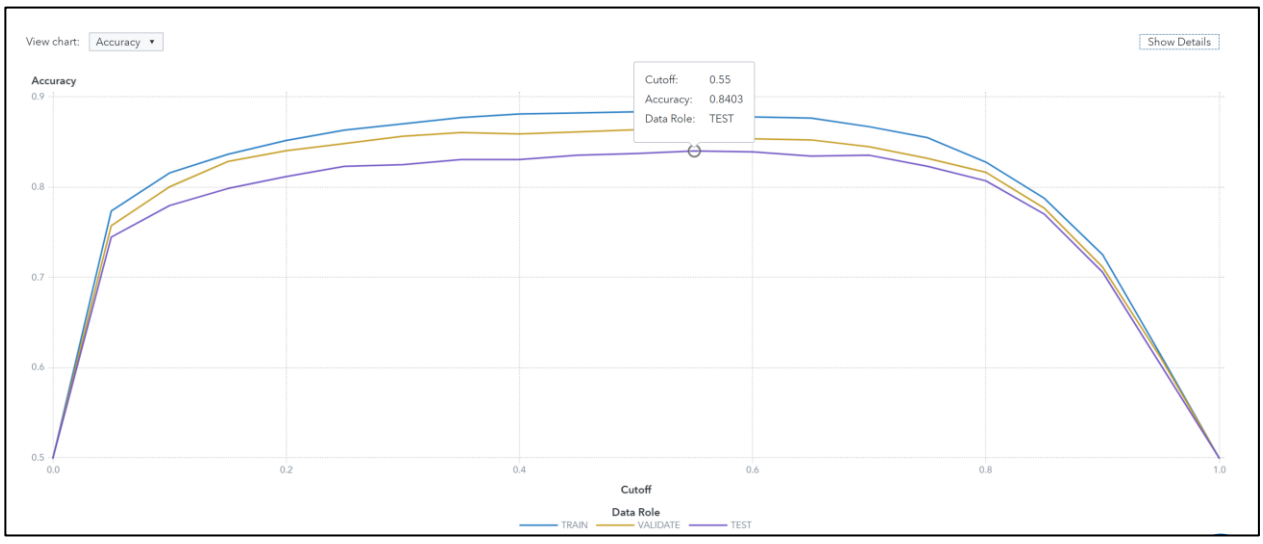


Figure 66: NN-10 Test Set Accuracy Threshold Curve

Analysing the accuracy curves for the 8 best models, reveals further interesting results. Figure 67 below highlights this, with a clear peak with RF-17's accuracy on the test set. This corroborates the values shown in Table X above, but also highlights the similarities in the majority of curves, conveying that most models achieve similar accuracies, given the same class threshold values. This is especially true for the three logistic regression models, as their accuracies are within the same percentile. Intriguingly SVM-22's accuracy is almost congruent with a normal distribution curve, with a sharp peak near the middle cutoff threshold of 0.5. This may suggest that the model is not as robust across cutoff thresholds, and instead relies on working towards a convergence point.

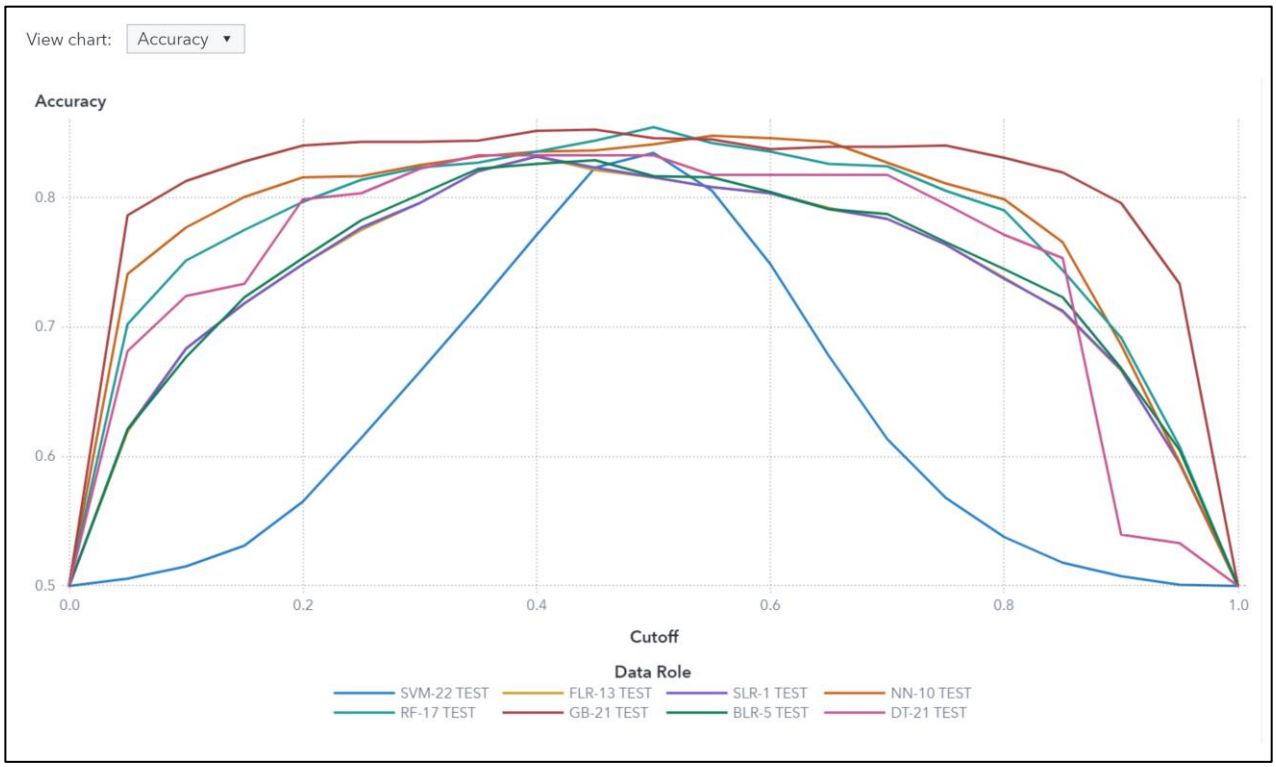


Figure 67: Accuracy curve for top performing models (test set)

Furthermore, plotted KS scores for the top models can be seen in Figure 68 below, again confirming the superiority of RF-17 (random forest). Another compelling insight is the distribution of attributes preferred by each top classifier. Figure 69 illustrates that contact, month and poutcome attributes were preferred by all eight models. This is very interesting, as it aligns with original predictions made within the data exploration section of the report, where cellular services may have been preferred by the bank due to a higher likelihood of answering the phone, as reported in section 4.2.5. Similarly, section 3.2.11 describes the distribution of month values, where it was originally mentioned that 'month' may be of great importance, especially given the time of data collection. As the BANK_DIRECT_MARKETING dataset was formulated during the global financial crisis, the models may have found key links between month and poutcome, which determined if clients previously signed up for a term deposit.

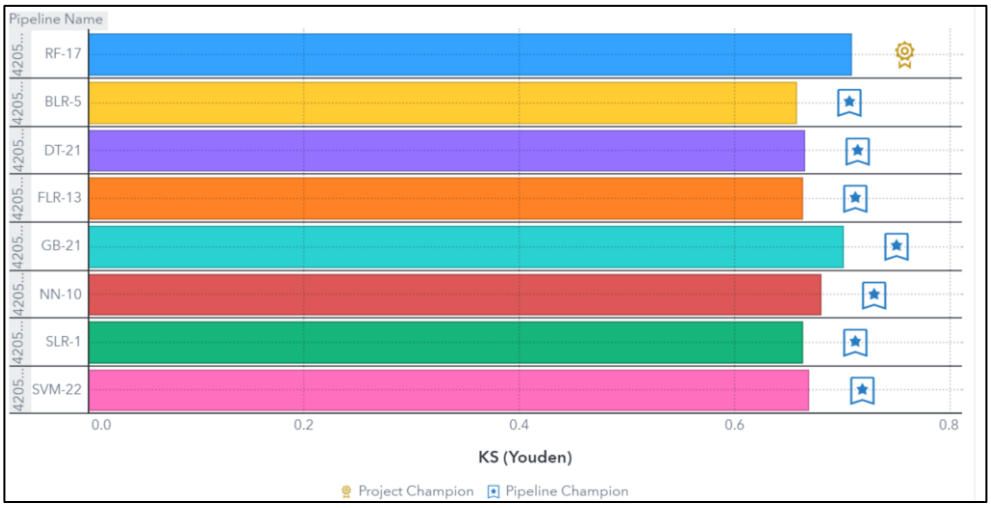


Figure 68: Bar chart – KS score for top performing models

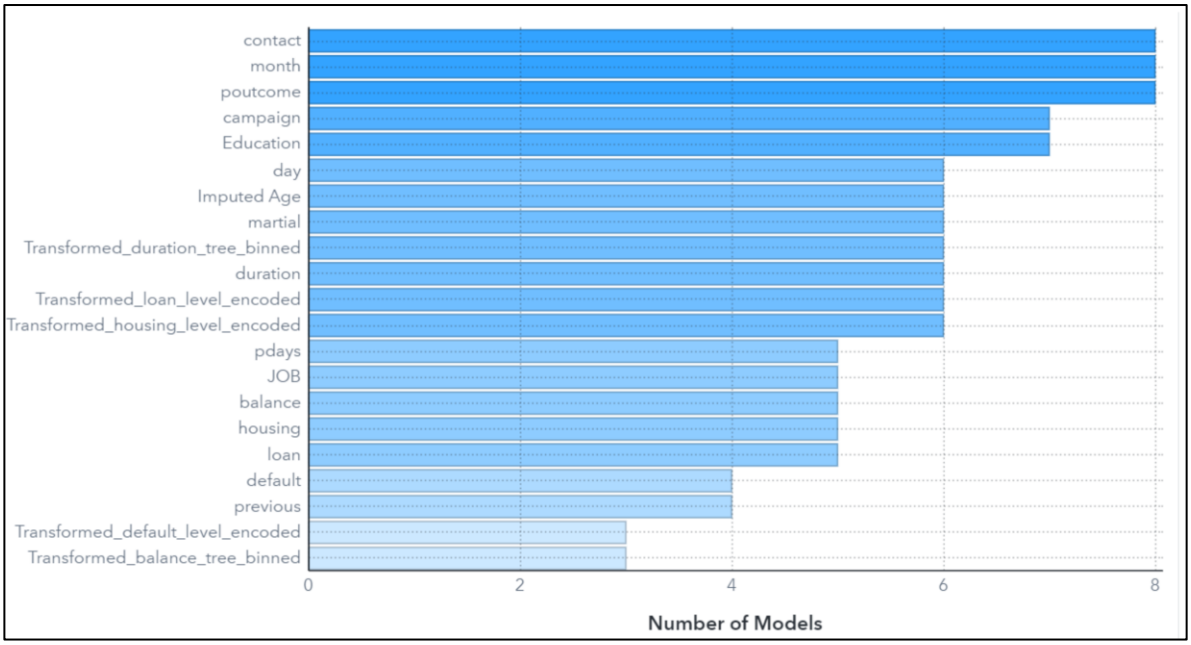


Figure 69: Bar chart – most preferred attributes by top performing models

Comparing this to the most important parameters for the champion model (RF-17) in Figure 70 below, it is evident that the random forest model prefers a high level of unique values, as present in attributes such as duration. With month being in second place, it may be assumed that the random forest is able to determine optimal splits depending on which month is most likely to have the longest call durations, and potentially a higher chance of term deposit registration.

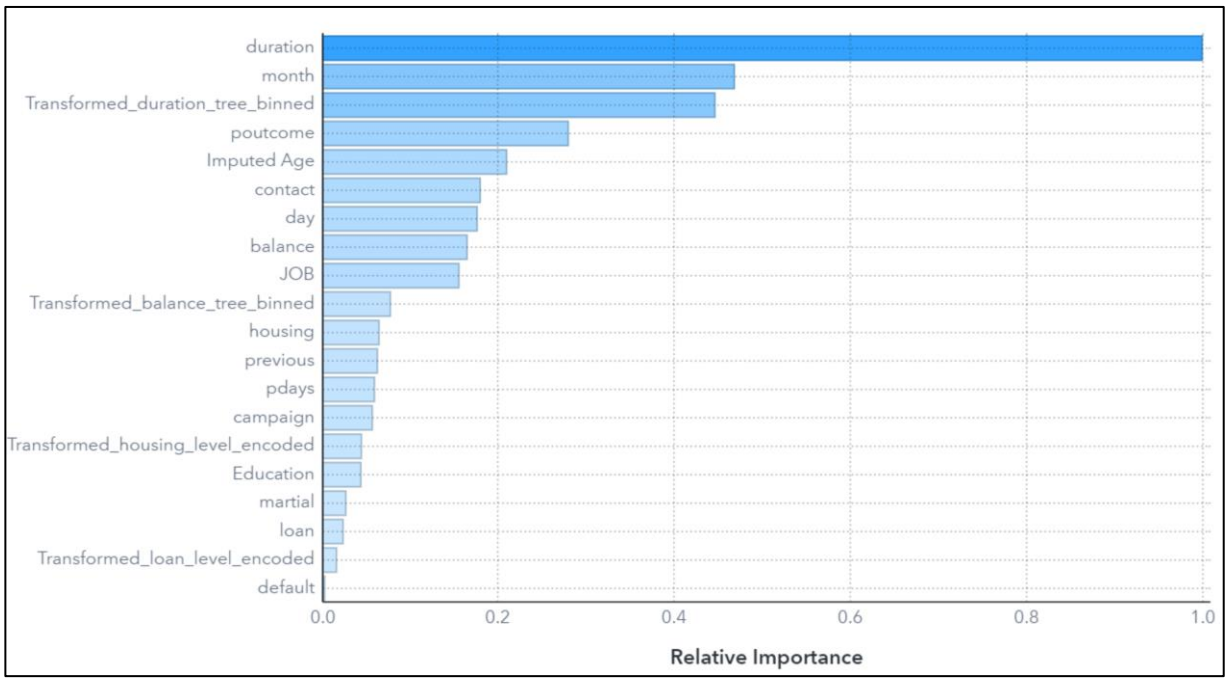


Figure 70: Bar chart – attributes preferred by champion model – RF17

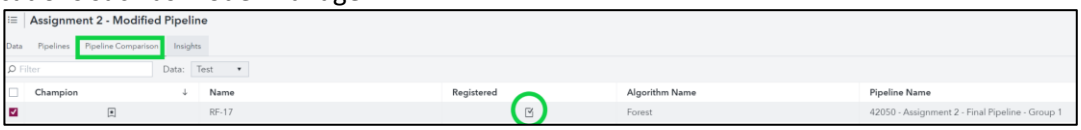
Following thorough comparative analysis, RF-17 (random forest) is declared as the champion model, exhibiting a KS score of 70.13%, and an accuracy of 85.07%. The second and third highest performing models are gradient boosted tree (GB-21) and neural network (NN-10), with all three being marginally different, with separations of less than a single accuracy percentage point.

Upon completion of comparative model assessment, it is imperative to know that homogeneity was considered with great importance at every stage of the data mining process. This involved utilising a homogenous preprocessing pipeline across all iterations, a constant seed value of 12345, at least 10 iterations per classifier to increase equal performance opportunities for models due to a lack of autotuning, and initialising projects with the same split for train (60%), validation (30%), and test (10%) data.

10. Model deployment

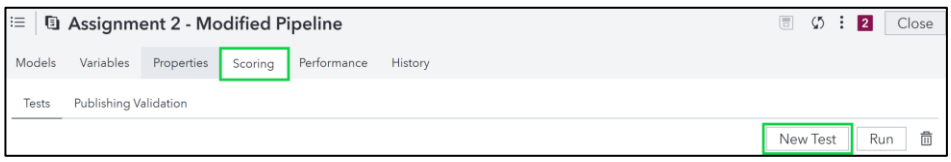
Model deployment is the final step in the data mining process, and when deployed, champion models must be regularly updated to fit on new input data. If champion models are not replaced for a while, they will begin to incorrectly predict new data, due to cumulative changes in real world data, that differs from its old training data. One common way to combat this is to regularly test a challenger model against the current champion model to identify which one has the best performance. If the challenger wins, it will replace the older deployed model. Adding to section 7.1.6 in the data mining process, we will go through the deployment process via SAS, beginning with steps to be carried out in Model Studio and Model Manager:

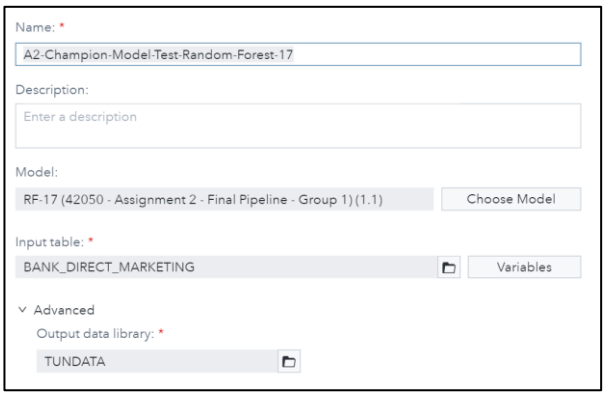
1. Champion model registration
   RF-17 was registered within the pipeline comparison tab in model studio to make it available to other SAS applications such as model manager.



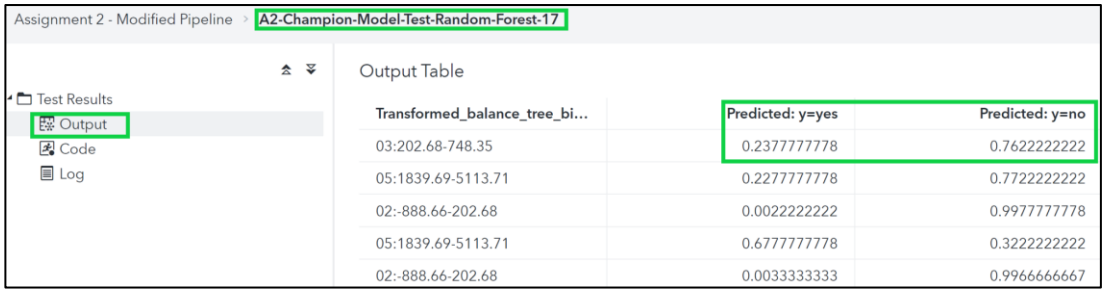
   Model Manager automatically selects ASTORE as the code type, which is suitable for random forest classifiers such as the champion model (RF-17). This can be seen below.

2. Open model manager to create a new scoring test for the champion model.
   A new test was created with the RF-17 model, with the correct input dataset selected (BANK_DIRECT_MARKETING). After saving this, the test can be run from the scoring tab.





3. View test outputs from the scoring tab
   As shown below, the champion model's predictions on the input table can be seen in the predicted yes and no columns. These are representative of the model's predictions on if a customer signed up for a term deposit or not.



11. Conclusion

After successfully completing the first iteration of our proposed data mining process, we were able to methodically analyse attribute types, distributions, and correlations, to provide a baseline for multivariate attribute comparisons. These investigations led to cluster discovery and formed precise viewpoints, leading to the formation of relationship narratives. An example of this is the connection between Pdays and Poutcome, where unknown outcomes indicate that the target client was never contacted before. This finding helped explain the enormous disproportionate distribution of 74.4% of all records being marked as 'unknown' for the Poutcome attribute. It was uncovered that all unknown values were directly linked to clients who were never contacted before ('-1'), as described by the Pdays attribute. This creates a strong holistic narrative that the Portuguese banking institution was skewed towards targeting a majority of new clients, instead of previously contacted clients. This decision could have been motivated by the global financial crisis at the time of data collection; and executed as a bid to increase their term deposit signup rate by introducing new targets.

Using these key insights, we proposed a robust pre-processing pipeline, which included key transformations such as level encoding to improve data readability and consistency, as well as tree-based binning, to optimally split attributes with large quantities of unique values, such as *balance* and *duration*. During pre-processing, we also suggested that homogeneity must be preserved where possible, and resultantly, implemented strategies to limit variation. Implementations such as constant random seeds, consistent pre-processing pipelines and completing at least 10 iterations per classifier, would yield the best possible chance to reach our target goal of at least 75-80% accuracy, with relatively good class separation (KS score).

Adhering to these same homogeneity principles, models were evaluated holistically on the same set of performance metrics, allowing for fair scrutiny across metrics such as accuracy, KS score, F1 score, ASE, AUC and SBC. Ultimately, the best performing model was RF-17, a random forest classifier, with an accuracy of 85.07% and a KS score of 70.13%. Following the final stage of our proposed data mining process, this model was then deployed and tested against the score dataset.

Overall, we successfully established the clear need for the implementation of our data mining strategy, and provided significant insights into relationships, observations, and nuances between model performance metrics, culminating to the selection of the most performant model. We also identified areas of future improvement such as in relation to class threshold modification across iterations, and cross validation loops. The inclusion of these in future projects may positively influence the model's class separation ability (KS score), and therefore generate a more robust model.

12. Challenges encountered during exploratory data analysis

Throughout the exploration stage of the analytics cycle, multiple challenges were faced, relating to the SAS environment itself, as well as the dataset. These are listed below with brief descriptions:

- Lack of clarity when using certain black box nodes such as anomaly detection
    - Specific nodes such as clustering and anomaly detection are not well documented, as they are majorly automated in nature. This made it difficult to gauge its results after running the pipeline. This was alleviated by researching on what the outputs mean in relation to the input dataset.

- Learner account features are restricted, and projects could not be shared
    - This created many issues for the team, essentially putting the final pipeline design on a single account, instead of being able to collaborate freely. Pipelines were also unable to be shared via the

exchange to other students for manual download. To resolve this issue, pipelines were manually uploaded and downloaded into a zip file for sharing. These were then re-uploaded on individual student accounts.

- Attribute binarization
    - To alter the default binarization from 1 and 2 to the more standard 0 and 1, a SAS code node was needed as it was not straightforward to complete this via the model studio user interface. We were able to resolve this issue by researching into how the code node works, and slightly tweaking the level encoding code to suit our needs.

- Scatter plot attribute type conversions
    - To display nominal / ordinal attributes on scatter plots, they had to be manually converted to their distinct count representations to plot against numerical values, which required significant research as this was not obvious at the time.

- Auto binning within SAS Visual Analytics
    - When viewing a continuous attribute with many levels, such as Age or Balance, SAS visual analytics auto bins to a maximum of 100. This made it hard to determine which values were either included or excluded within each bin, as these properties were not explicit. External software tools were used to achieve higher granularity where needed.

External factors were also found to marginally impact performance metrics throughout the iterative model assessment process. After intensive scrutiny and parameter changes, it became evident that identical configurations of nodes produced marginally varying results when re-run after pipeline sharing. This may be due to hidden randomised weights or random initialisation points in classifiers, or due to SAS' parallel cloud compute architecture. Attributable to the stochastic nature of this external phenomenon, little could be done to prevent this, which should be considered when re-running the final pipeline. During model training, this was minimised from occurring by running each node individually instead of with the entire pipeline. This alludes more to limitations with SAS' parallel model training cloud architecture, perhaps due to learner licence restrictions. Nevertheless, it is **more likely, and also completely feasible** that this occurred due to the inherent non-deterministic nature of all classifiers. This is due to each training iteration being slightly different than the one prior, irrespective of common configurations.

13. Appendix

13.1. Neural Network

13.1.1. Neural Network Configuration Table

NN Iteration Parameters												
Iteration No.	**No. of Hidden Layers**	**Neurons per Hidden Layer**	**Hidden Layer Activation Function**	**Number of Tries**	**Max No. of Iterations**	**L1**	**L2**	**Accuracy**	**KS**	**F1**	**ASE**	**AUC**
NN-1	1	50	Tanh	1	300	0	0.1	0.8185	0.6465	0.8192	0.1289	0.8955
NN-2	1	32	Tanh	1	300	0	0.1	0.8242	0.6484	0.8184	0.1282	0.8948
NN-3	1	32	Tanh	1	300	0	0.001	0.5	0	0.6667	0.25	0.5
NN-4	1	32	Relu	1	300	0	0.001	0.8318	0.6635	0.8241	0.1224	0.9037
NN-5	1	32	Tanh	1	300	0.001	0.001	0.8242	0.6522	0.8229	0.1256	0.8993
NN-6	1	100	Relu	1	500	0	0.01	0.8299	0.6597	0.8235	0.1257	0.8961
NN-7	1	100	Relu	1	600	0	0.01	0.8327	0.6692	0.8242	0.1222	0.9033
NN-8	2	100	Tanh	1	400	0	0.01	0.8204	0.6408	0.8177	0.1285	0.8941
NN-9	2	100	ReLU	1	600	0.0001	0.01	0.5	0	0	0.25	0.5
#1 NN-10	2	100	ReLu	1	600	0	0.01	0.8374	0.6805	0.8290	0.1177	0.9097
NN-11	2	100	Tanh	1	600	0	0.01	0.8251	0.6503	0.8174	0.1277	0.8941
NN-12	3	100/50	ReLu/Tanh	1	600/300	0	0.01	0.5	0	0	0.25	0.5
NN-13	1/2	150	Relu/Tanh	1	600/300	0	0.01	NA	NA	NA	NA	NA
NN-14	2 / 1	100/50	Relu/Tanh	2	700	0	0.01	NA	NA	NA	NA	NA

13.2. Logistic Regression

13.2.1. Logistic regression parameter descriptions

Logistic Regression Parameter	**Description**
Effect Selection Criterion *Criterion to determine the order independent variables enter or leave at each step.*	AIC (Akaike Information Criterion): penalises the complexity of the model based on no. of coefficients and rewards goodness of fit. Lower values are better. AICC (Corrected Akaike Information Criterion): A correction to the AIC, taking sample size into consideration (typically used for smaller samples). SBC (BIC) (Schwarz Bayesian Criterion): penalises model complexity/high dimensionality more than AIC. Lower values indicate better models.

Default: SBC (BIC)	Significance level: threshold at which effects are deemed statistically significant and thus included in the model.
Selection Process Stopping Criterion *Criterion to determine when to halt the regression process. Default: SBC (BIC)*	Selection stops when the following criterion start increasing: AIC, AICC, Validation Set (Average Squared Error), SBC (BIC), Significance level (default: 0.05) Validation ASE (Average Squared Error): average of squares of the errors between the predicted and observed values.
Max No. of Effects *Default: 0 (no max limit)*	The maximum number of predictor variables (effects) that will be allowed in the model at a given step. This parameter controls model complexity, and by increasing this value, we explore the trade-off between model simplicity and model fit.
Max No. of Steps *Default: 0 (option is ignored)*	Specifies the max no. of steps to complete during regression. This will halt regression training even if stopping criterion is not yet met.
Optimisation Technique	Different techniques may converge faster or be more stable for specific types of data. Options include: None, Conjugate-gradient, Double-dogleg, Dual quasi-Newton, Nelder-Mead simplex, Newton-Raphson, Newton-Raphson w/ ridging, Trust-region.
Max No. of Iterations *Default: blank/NA*	By increasing this number, more optimisation refinement can occur for the selected optimisation technique, but at the cost of computational load. Each technique has a predefined iteration count when left blank (SAS, n.d.).
Absolute function convergence	A small value that determines the forced stopping point irrespective of training.

13.2.2. FLR Configuration Tables

FLR Iteration Parameters							
Iteration No.	**Effect Selection Criterion**	**Selection Process Stopping Criterion**	**Max No. of Effects**	**Max No. of Steps**	**Optimisation Technique**	**Max No. of Iterations**	**Absolute function convergence**
FLR-1	AIC	AIC	5	10	Conjugate-gradient	10	0.0001
FLR-2	AICC	AICC	6	12	Newton-Raphson	20	0.0002
FLR-3	SBC(BIC)	Validation ASE	7	13	Newton-Raphson with ridging	35	0.0025
FLR-4	Significance level	Validation ASE	8	15	Newton-Raphson with ridging	40	0.0004
FLR-5	AIC	SBC(BIC)	10	18	Conjugate-gradient	60	0.0005
FLR-6	AICC	AIC	10	18	Trust-region	60	0.0006
FLR-7	SBC(BIC)	SBC(BIC)	12	21	Trust-region	70	0.0007
FLR-8	Significance level	Significance level	12	22	Newton-Raphson	80	0.0008
FLR-9	AIC	Validation ASE	14	26	Trust-region	90	0.0009

FLR-10	AICC	SBC(BIC)	14	25	Newton-Raphson with ridging	100	0.0010
FLR-11	AIC	AICC	15	27	Dual quasi-Newton	110	0.0011
FLR-12	AICC	Significance level	16	29	Trust-region	120	0.0012
FLR-13	SBC(BIC)	Validation ASE	17	30	Conjugate-gradient	130	0.0013
FLR-14	Significance level	AIC	18	32	Newton-Raphson	140	0.0014
FLR-15	AIC	Significance level	19	34	Conjugate-gradient	150	0.0015
FLR-16	AICC	AICC	20	35	Newton-Raphson with ridging	160	0.0016
FLR-17	SBC(BIC)	Significance level	21	37	Dual quasi-Newton	170	0.0017
FLR-18	Significance level	Validation ASE	22	39	Trust-region	180	0.0018
FLR-19	AIC	AICC	23	40	Conjugate-gradient	190	0.0019
FLR-20	AICC	SBC(BIC)	24	42	Newton-Raphson	200	0.0020
FLR-21	SBC(BIC)	SBC(BIC)	-	-	Newton-Raphson with ridging	-	-
FLR-22	SBC(BIC)	SBC(BIC)	-	-	Newton-Raphson with ridging	-	-
FLR-23	SBC(BIC)	SBC(BIC)	12	21	Trust-region	70	0.0007

FLR Iteration Results (Test Set)					
Iteration No.	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under the Curve**
FLR-1	0.8119	0.6238	0.8124	0.1321	0.8921
FLR-2	0.8138	0.6578	0.8154	0.1303	0.8931
FLR-3	0.8214	0.6560	0.8235	0.1298	0.8939
FLR-4	0.8176	0.6503	0.8181	0.1300	0.8928
FLR-5	0.8138	0.6578	0.8154	0.1299	0.8941
FLR-6	0.8138	0.6578	0.8154	0.1303	0.8931
FLR-7	0.8129	0.6635	0.8136	0.1286	0.8948
FLR-8	0.8176	0.6503	0.8181	0.1300	0.8928
FLR-9	0.8138	0.6578	0.8154	0.1303	0.8931
FLR-10	0.8138	0.6578	0.8154	0.1303	0.8931
FLR-11	0.8138	0.6578	0.8154	0.1303	0.8932
FLR-12	0.8138	0.6578	0.8154	0.1303	0.8931
#1 FLR-13	0.8119	0.6635	0.8124	0.1283	0.8959
FLR-14	0.8176	0.6503	0.8181	0.1300	0.8928
FLR-15	0.8138	0.6578	0.8154	0.1299	0.8941
FLR-16	0.8138	0.6578	0.8154	0.1303	0.8931
FLR-17	0.8119	0.6635	0.8124	0.1286	0.8948
FLR-18	0.8176	0.6503	0.8181	0.1300	0.8928
FLR-19	0.8138	0.6578	0.8154	0.1299	0.8941
FLR-20	0.8138	0.6578	0.8154	0.1303	0.8931

FLR-21	0.8129	0.6635	0.8136	0.1286	0.8948
FLR-22	0.8072	0.6371	0.8046	0.1323	0.8919
FLR-23	0.8072	0.6446	0.8050	0.1334	0.8907

13.2.3. BLR Configuration Tables

BLR Iteration Parameters							
Iteration No.	Effect Selection Criterion	Selection Process Stopping Criterion	Min No. of Effects	Max No. of Steps	Optimisation Technique	Max No. of Iterations	Absolute function convergence
BLR-1 (Default)	SBC (BIC)	SBC (BIC)	0	0	Newton-Raphson with ridging	NA (50)	NA
BLR-2	AIC	SBC (BIC)	15	0	Double dogleg	NA (200)	NA
BLR-3	Significance level	Validation ASE	10	0	Conjugate-gradient	NA (400)	NA
BLR-4	AICC	Significance level	12	0	Newton-Raphson with ridging	150	0.00000001
BLR-5	SBC (BIC)	Validation ASE	0	0	Trust-region	95	NA
BLR-6	AIC	AICC	16	0	Nelder-Mead simplex	2000	0.00000005
BLR-7	SBC (BIC)	SBC (BIC)	9	0	Newton-Raphson with ridging	NA (50)	NA
BLR-8	AICC	SBC (BIC)	0	0	Dual quasi-Newton	250	0.00000002
BLR-9	Significance level	AICC	5	0	Newton-Raphson with ridging	85	0.0000003
BLR-10	SBC (BIC)	AIC	5	0	Trust-region	NA (50)	0.00000001
BLR-11 (Default prior to binning)	SBC (BIC)	SBC (BIC)	0	0	Newton-Raphson with ridging	NA (50)	NA
BLR-12 (Best model BLR-5, prior to binning)	SBC (BIC)	Validation ASE	0	0	Trust-region	95	NA

BLR Iteration Results (Test Set)					
Iteration No.	Accuracy	KS score	F1 score	Average Squared Error	Area Under the Curve
BLR-1 (Default)	0.8129	0.6578	0.8132	0.1282	0.8957
BLR-2	0.8204	0.6541	0.8214	0.1289	0.8952
BLR-3	0.8147	0.6314	0.8168	0.1308	0.8931
BLR-4	0.8157	0.6465	0.8162	0.1307	0.8911
#1 BLR-5	0.8166	0.6578	0.8173	0.1279	0.8969
BLR-6	0.8176	0.6446	0.8188	0.1303	0.8913
BLR-7	0.8129	0.6578	0.8132	0.1282	0.8957
BLR-8	0.8138	0.6578	0.8154	0.1303	0.8932
BLR-9	0.8157	0.6389	0.8176	0.1321	0.8889
BLR-10	0.8166	0.6578	0.8173	0.1279	0.8969
BLR-11 (Default prior to binning)	0.8072	0.6371	0.8046	0.1323	0.8919
BLR-12	0.8100	0.6484	0.8069	0.1310	0.8933

(Best model BLR-5, prior to binning)					

13.2.4. SLR Configuration Tables

SLR Iteration Parameters							
Iteration No.	**Effect Selection Criterion**	**Selection Process Stopping Criterion**	**Max No. of Effects**	**Max No. of Steps**	**Optimisation Technique**	**Max No. of Iterations**	**Absolute function convergence**
SLR-1 (Default)	SBC (BIC)	SBC (BIC)	0	0	Newton-Raphson with ridging	NA (50)	NA
SLR-2	AIC	SBC (BIC)	15	0	Double dogleg	NA (200)	NA
SLR-3	Significance level	Validation ASE	10	0	Conjugate-gradient	NA (400)	NA
SLR-4	AICC	Significance level	12	0	Newton-Raphson with ridging	150	0.00000001
SLR-5	SBC (BIC)	Validation ASE	0	0	Trust-region	95	NA
SLR-6	AIC	AICC	16	0	Nelder-Mead simplex	2000	0.00000005
SLR-7	SBC (BIC)	SBC (BIC)	9	0	Newton-Raphson with ridging	NA (50)	NA
SLR-8	AICC	SBC (BIC)	0	0	Dual quasi-Newton	250	0.00000002
SLR-9	Significance level	AICC	5	0	Newton-Raphson with ridging	85	0.0000003
SLR-10	SBC (BIC)	AIC	5	0	Trust-region	NA (50)	0.00000001
SLR-11 (Default prior to binning)	SBC (BIC)	SBC (BIC)	0	0	Newton-Raphson with ridging	NA (50)	NA
SLR-12 (Best model SLR-5, prior to binning)	SBC (BIC)	Validation ASE	0	0	Trust-region	95	NA

SLR Iteration Results (Test Set)					
Iteration No.	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under the Curve**
#1 SLR-1 (Default)	0.8157	0.6635	0.8176	0.1299	0.8930
SLR-2	0.8138	0.6578	0.8154	0.1303	0.8931
SLR-3	0.8176	0.6522	0.8181	0.1299	0.8934
SLR-4	0.8138	0.6578	0.8154	0.1303	0.8931
SLR-5	0.8157	0.6635	0.8176	0.1299	0.8930
SLR-6	0.8138	0.6578	0.8154	0.1300	0.8945
SLR-7	0.8157	0.6635	0.8176	0.1299	0.8930
SLR-8	0.8138	0.6578	0.8154	0.1303	0.8932
SLR-9	0.8119	0.6238	0.8124	0.1325	0.8906
SLR-10	0.8157	0.6635	0.8176	0.1299	0.8930
SLR-11 (Default prior to binning)	0.8072	0.6371	0.8046	0.1323	0.8919
SLR-12	0.8072	0.6371	0.8046	0.1323	0.8919

(Best model SLR-1, prior to binning)					

13.3. Decision Tree

13.3.1. Decision Tree parameter descriptions

Potential Parameters Changes Influencing the Outcome:

- Grow Criterion: This function is used to measure the quality of a split. Choices considered here include "Information Gain Ratio", "Gini", and "Entropy". The choice here can influence how the tree prioritises splits.
    - Information Gain Ratio: A modification of information gain which considers the number of branches that would result after the split. It adjusts the gain for splits that result in a larger number of branches.
    - Gini: A measure of impurity or disorder. A Gini score of 0 represents perfect classification, while higher values indicate more disordered classifications.
    - Entropy: Represents disorder or impurity in the dataset. The tree aims to reduce entropy with each split.

- Interval Target Criterion: Determines the criterion for splitting continuous target variables. Options here include "Variance", "F-Test", and "CHAID".
    - Variance: Splits are based on variance reduction
    - F-Test: Utilizes the F-Test to decide upon the best split. This tests the variance between the two sets of data.
    - Chaid: A type of decision tree algorithm that uses the Chi-squared test to determine the best split.

- Bonferroni: This is a statistical method used to adjust for multiple comparisons, ensuring that the significance level is met across all tests and not just one.
    - NA: Not applicable or not used
    - Disabled: Bonferroni correction is not applied
    - Enabled: Bonferroni correction is applied

Maximum number of branches: This specifies the maximum number of branches or outcomes which can emerge from a single decision node. It is used to control the granularity and complexity of the tree.

Maximum Depth: This determines how deep the tree can be. Given our large dataset, setting a fitting max depth can help with preventing overfitting as we have multiple attributes.

Minimum Leaf Size: This indicates the minimum samples a leaf node can have. Low setting might capture noise in the data, especially if a category has few samples (for example, rare job types).

13.3.2. DT Configuration Tables

DT Iteration Parameters						
Iteration No.	**Class Target Criterion**	**Interval Target Criterion**	**Bonferroni**	**Max. No of Branches**	**Max. Depth**	**Min. Leaf Size**
DT-1 (Default)	IGR	Variance	NA	2	10	5
DT-2	Gini	Variance	NA	2	10	5
DT-3	Entropy	Variance	NA	2	10	5
DT-4	IGR	F Test	Disabled	2	10	5
DT-5	Gini	F Test	Disabled	2	10	5
DT-6	Entropy	F Test	Disabled	2	10	5
DT-7	IGR	F Test	Enabled	2	10	5
DT-8	Gini	F Test	Enabled	2	10	5
DT-9	Entropy	F Test	Enabled	2	10	5
DT-10	IGR	CHAID	Enabled	2	10	5
DT-11	Gini	CHAID	Enabled	2	10	5

DT-12	Entropy	CHAID	Enabled	2	10	5
DT-13	IGR	CHAID	Disabled	2	10	5
DT-14	Gini	CHAID	Disabled	2	10	5
DT-15	Entropy	CHAID	Disabled	2	10	5
DT-16	Entropy	F Test	Disabled	10	30	5
DT-17	IGR	CHAID	Disabled	10	30	5
DT-18	Entropy	CHAID	Disabled	2	10	10
DT-19	Entropy	F Test	Disabled	2	10	10
DT-20	IGR	CHAID	Disabled	10	30	10
DT-21 (Best model DT-15, prior to binning)	Entropy	CHAID	Disabled	2	10	5

DT Iteration Results (Test Set)					
Iteration No.	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under the Curve**
DT-1 (Default)	0.8119	0.6238	0.8142	0.14424	0.8591
DT-2	0.8166	0.6408	0.8262	0.13857	0.8712
DT-3	0.8308	0.6616	0.8350	0.13074	0.8812
DT-4	0.8119	0.6238	0.8142	0.14424	0.8591
DT-5	0.8166	0.6408	0.8262	0.13857	0.8712
DT-6	0.8308	0.6616	0.8350	0.13074	0.8812
DT-7	0.8119	0.6238	0.8142	0.14424	0.8591
DT-8	0.8166	0.6408	0.8262	0.13857	0.8712
DT-9	0.8308	0.6616	0.8350	0.13074	0.8812
DT-10	0.8119	0.6238	0.8142	0.14424	0.8591
DT-11	0.8166	0.6408	0.8262	0.13857	0.8712
DT-12	0.8308	0.6616	0.8350	0.13074	0.8812
DT-13	0.8119	0.6238	0.8142	0.14424	0.8591
DT-14	0.8166	0.6408	0.8262	0.13857	0.8712
DT-15	0.8308	0.6616	0.8350	0.13074	0.8812
DT-16	0.8166	0.6484	0.8233	0.13872	0.8769
DT-17	0.8233	0.6484	0.8302	0.14028	0.8642
DT-18	0.8270	0.6578	0.8307	0.13103	0.8814
DT-19	0.8270	0.6578	0.8307	0.13103	0.8814
DT-20	0.8289	0.6578	0.8362	0.13355	0.8718
#1 DT-21 (Best model DT-15, prior to binning)	0.8327	0.6654	0.8395	0.1301	0.8819

13.4. Gradient Boosted

13.4.1. Gradient Boosted Tree parameter descriptions

Iteration No.: This represents the sequence number of each run or test. It helps in tracking the progress and comparing the performance of different iterations.

Number of Trees: This is the number of boosting stages to be run. In gradient boosting, a sequence of decision trees is built. Increasing the number of trees can increase the model's performance, but after a certain point, it might lead to overfitting or longer training times.

Learning Rate: Also known as shrinkage or step size, the learning rate scales the contribution of each tree added to the model. A smaller value can lead to a more robust model by shrinking the feature weights to make the boosting process more conservative.

Subsample Rate: This represents the fraction of samples used for fitting the individual base learners. If it's set to a value less than 1.0, it introduces randomness into the model, making the training more robust to noise.

L1 Regularization: Known as Lasso Regression, L1 regularization adds a penalty equivalent to the absolute value of the magnitude of coefficients. This can lead to some coefficients becoming zero, effectively leading to feature selection.

L2 Regularization: Referred to as Ridge Regression, L2 regularization adds a penalty equivalent to the square of the magnitude of coefficients. It prevents overfitting by discouraging overly complex models which have coefficients that are too large.

KS Score: The Kolmogorov-Smirnov score gauges the model's discriminatory power. A high KS Score means the model does an excellent job distinguishing between the positive and negative classes.

F1 Score: The F1 Score is the harmonic mean of precision and recall. It's especially useful when class distribution is imbalanced. A higher F1 Score signifies a better balance between precision and recall.

Accuracy: This metric provides the ratio of correctly predicted instances to the total instances. A higher accuracy indicates a better-performing model, but it's crucial to consider it alongside other metrics, especially in imbalanced datasets.

Average Squared Error (ASE): It represents the average of the squares of the differences between the predicted and actual values. Lower values of ASE indicate a model that predicts closer to the actual values.

Area Under ROC (AUC): The Receiver Operating Characteristic (ROC) curve is a graphical representation of the true positive rate against the false positive rate. The AUC measures the entire two-dimensional area underneath the entire ROC curve. A value of 1 represents a perfect model, and a value of 0.5 indicates a model no better than random guessing. Explanation of Parameters/Columns:

13.4.2. GB Configuration Table

GB Iteration Parameters										
Iteration No.	**No. of Trees**	**Learning rate**	**Subsample rate**	**L1 regularisation**	**L2 regularisation**	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under Curve**
GB-1 (Default)	100	0.01	0.5	0	1	0.8346	0.6843	0.8405	0.1357	0.9041
GB-2	110	0.02	0.6	0.5	0.8	0.8374	0.6843	0.8434	0.1248	0.9067
GB-3	90	0.03	0.7	1	0.6	0.8393	0.6805	0.8452	0.1152	0.9087
GB-4	105	0.015	0.55	1.5	0.4	0.8384	0.6824	0.8444	0.1184	0.9061
GB-5	115	0.025	0.65	2	0.285	0.8393	0.6805	0.8455	0.1334	0.9023
GB-6	85	0.035	0.75	2.5	1.2	0.8440	0.6881	0.8496	0.1151	0.9095
GB-7	95	0.04	0.8	0.25	1.4	0.8440	0.6881	0.8501	0.1160	0.9092
GB-8	120	0.045	0.85	1.75	1.6	0.8403	0.6862	0.8473	0.1141	0.9089
GB-9	125	0.05	0.9	0.75	0.9	0.8374	0.6805	0.8442	0.1172	0.9091
GB-10	80	0.055	0.95	2.25	0.7	0.8459	0.6919	0.8514	0.1147	0.9079
GB-11	130	0.06	0.5	1.25	0.5	0.8384	0.6786	0.8438	0.1192	0.9067
GB-12	135	0.065	0.6	0.5	0.3	0.8422	0.6881	0.8478	0.1184	0.9062
GB-13	75	0.07	0.7	3	1.1	0.8403	0.6805	0.8462	0.1156	0.9055
GB-14	140	0.075	0.8	2.75	1.3	0.8355	0.6881	0.8418	0.1152	0.9105
GB-15	145	0.08	0.85	1.5	1.5	0.8393	0.6862	0.8466	0.1165	0.9058
GB-16	150	0.085	0.9	0.25	1.7	0.8412	0.6919	0.8470	0.1174	0.9064
GB-17	70	0.09	0.95	2.5	1.9	0.8422	0.6843	0.8480	0.1150	0.9091
GB-18	200	0.075	0.1	0.5	1	0.8261	0.6616	0.8321	0.1258	0.8997
GB-19	160	0.1	0.6	1	0.6	0.8365	0.6824	0.8434	0.1164	0.9088
GB-20 (Default – without binning)	100	0.01	0.5	0	1	0.8393	0.6824	0.8463	0.1352	0.9023
#1 GB-21	150	0.085	0.9	0.25	1.7	0.8478	0.6957	0.8546	0.1174	0.9087

(Best model - prior to binning)										

13.5. Random Forest

13.5.1. Random Forest parameter descriptions

Random Forest Parameter	Description
No. of Trees *Default: 100*	Number of trees in the random forest ensemble
Class Target Voting Method *Default: Probability*	Probability – averages all leaf node probabilities across trees to determine a final class prediction (Dataiku, 2023). Majority – measures the proportion of trees that predicted the same class (most occurring). Each tree's leaf node has a proportion % for yes and no values to determine the tree's outcome (e.g., 80% of values in a tree's leaf node is 'yes').
Class Target Criterion *Default: Info Gain Ratio*	This specifies the best splits based on the target variable. Split criterion include: CHAID, Chi-square, Entropy, Gini, Info Gain Ratio.
Max. No. of Branches *Default: 2*	This value determines the number of branches to consider in the hierarchy prior to a node split. Ranges from 2-5.
Max. Depth *Default: 20*	Specifies the max vertical depth for each tree in the forest. Ranges from 1-50.
Leaf Size Specification *Default: Count*	Determines the method used to select the min. leaf size. Can be based on count of all observations or proportion.
Min. Leaf Size (used when COUNT is selected for leaf size specification) *Default: 5*	Smallest No. of observation present in a new branch.
Min. Leaf Proportion (used when PROPORTION is selected for leaf size specification) *Default: 0.00005*	Smallest fractional proportion of observations with an available target value
Interval Bin No. *Default: 50*	Number of bins
Binning Method *Default: Quantile*	Bucket – divide input attributes into equally spaced intervals based on max and min value differences. Quantile – divide input attributes into equally sized groups based on percentage bounds.
In-bag sample proportion *Default: 0.6*	Specify the proportion/subset of the training data to train a tree with.

13.5.2. RF Configuration Tables

RF Iteration Parameters											
Iteration No.	No. of Trees	Class Target Voting Method	Class Target Criterion	Max. No. of Branches	Max. Depth	Leaf Size Spec	Min. Leaf Size	Min. Leaf Proportion	Interval Bin No.	Bin Method	In-bag proportion
RF-1 Default	100	Probability	Info Gain	2	20	Count	5	NA	50	Quantile	0.6
RF-2	250	Probability	Info Gain	4	35	Count	5	NA	50	Quantile	0.6
RF-3	300	Majority	Gini	3	20	Count	5	NA	35	Quantile	0.6
RF-4	250	Majority	CHAID	4	30	Count	6	NA	45	Quantile	0.5

RF-5	550	Majority	Gini	2	20	Count	3	NA	50	Quantile	0.5
RF-6	700	Probability	Entropy	3	40	Count	5	NA	55	Quantile	0.4
RF-7	300	Probability	Entropy	5	50	Proportion	NA	0.00003	20	Bucket	0.6
RF-8	850	Majority	Chi-Square	5	20	Count	10	NA	50	Quantile	0.6
RF-9	1000	Majority	Info Gain	2	20	Count	5	NA	50	Quantile	0.6
RF-10	1000	Probability	Info Gain	2	20	Count	5	NA	50	Quantile	0.6
RF-11	675	Majority	Gini	3	25	Count	6	NA	40	Bucket	0.6
RF-12	900	Probability	CHAID	2	45	Proportion	NA	0.00007	25	Quantile	0.6
RF-13 Default (before binning)	100	Probability	Info Gain	2	20	Count	5	NA	50	Quantile	0.6
RF-14	50	Probability	Chi-Square	5	10	Count	20	NA	100	Bucket	0.5
RF-15	85	Majority	Info Gain	4	5	Count	5	NA	30	Quantile	0.6
RF-16	950	Probability	Info Gain	5	35	Count	5	NA	50	Quantile	0.7
RF-17	900	Majority	Info Gain	5	35	Count	5	NA	50	Quantile	0.7
RF-18	750	Probability	Info Gain	3	50	Count	4	NA	50	Quantile	0.4
RF-19	925	Probability	Info Gain	5	45	Count	5	NA	55	Quantile	0.8
RF-20 (Best Model config, but before tree binning)	900	Majority	Info Gain	5	35	Count	5	NA	50	Quantile	0.7
RF-21 (Best model, but with bucket binning instead)	900	Majority	Info Gain	5	35	Count	5	NA	50	Bucket	0.7

Random Forest Iteration Results (Test Set)					
Iteration No.	Accuracy	KS score	F1 score	Average Squared Error	Area Under the Curve
RF-1	0.8355	0.6843	0.8404	0.1187	0.9035

Default					
RF-2	0.8488	0.6975	0.8548	0.1158	0.9083
RF-3	0.8374	0.6786	0.8453	0.1159	0.9110
Worst - RF-4	0.7164	0.4329	0.7217	0.1865	0.7986
RF-5	0.8450	0.6900	0.8523	0.1151	0.9097
RF-6	0.8384	0.6805	0.8447	0.1180	0.9116
RF-7	0.8393	0.6881	0.8446	0.1177	0.9090
RF-8	0.8393	0.6786	0.8468	0.1167	0.9094
RF-9	0.8393	0.6862	0.8449	0.1185	0.9052
RF-10	0.8384	0.6824	0.8438	0.1168	0.9074
RF-11	0.8440	0.6881	0.8523	0.1157	0.9113
RF-12	0.7703	0.5463	0.7692	0.1655	0.8481
RF-13 Default (before binning)	0.8431	0.6900	0.8488	0.1166	0.9086
RF-14	0.8318	0.6635	0.8373	0.1260	0.9047
RF-15	0.8043	0.6635	0.7973	0.1340	0.8933
#2 RF-16	0.8497	0.6994	0.8558	0.1151	0.9108
#1 RF-17	0.8507	0.7013	0.8577	0.1152	0.9096
RF-18	0.8431	0.6994	0.8485	0.1160	0.9081
#3 RF-19	0.8497	0.6994	0.8558	0.1152	0.9089
RF-20 (#17 Best Model config, but before tree binning)	0.8488	0.6975	0.8553	0.1142	0.9122
RF-21 (Best model, but with bucket binning instead)	0.8412	0.6862	0.8486	0.1161	0.9076

13.6. Support Vector Machine

13.6.1. SVM Parameter descriptions

There are a few reasons for changing the parameter in SVM:

- Penalty: miss classifications errors
  - Lowering the penalty (0.1) gives the model a broader boundary, allowing a better generalisation of data and preventing overfitting.
  - A moderate penalty (1) would increase the accuracy but reduce the margin, which results in a lower KS.
  - A very high penalty (>10) significantly enhances the accuracy but will expose the model to overfitting issues, consequently reducing the KS score.
- Kernel: Math function to transform data in higher dimension
  - Linear Kernal is compatible with a simple model where the data can be divided by a simple line. The model is simple and requires minimal computer resources, but it cannot work with non-linear or high-dimensional data.
  - Polynomial Kernal is very suitable for learning and predicting polynomial patterns.
- Tolerance: balances the number of support vectors and models accuracy
  - A low tolerance range like the default setting (0.000001) can increase the accuracy but require more computer resources and prompt overfitting.
  - A higher range will fail to capture complex and high dimensional datasets but demand longer training time. Ideally, the model will have a low penalty (0.1), Poly Kernel and moderate tolerance.

13.6.2. SVM Configuration Table

SVM Iteration Parameters									
Iteration No.	**Penalty**	**Kernel**	**Tolerance**	**Max iteration**	**Accuracy**	**KS score**	**F1 score**	**Average Squared Error**	**Area Under Curve**
SVM-1 (Default, prior to binning)	1	Linear	0.000001	25	0.8204	0.6408	0.8201	0.1861	0.8860
SVM-2 (Default)	1	Linear	0.000001	25	0.8185	0.6371	0.8157	0.1687	0.8937
SVM-3	1	Linear	0.0001	25	0.8176	0.6352	0.8150	0.1687	0.8936
SVM-4	0.1	Poly (2)	0.6	10	0.8280	0.6616	0.8240	0.1564	0.9029
SVM-5	0.1	Linear	0.0001	25	0.8157	0.6408	0.8138	0.1655	0.8945
SVM-6	0.1	Poly (2)	0.0001	25	0.8318	0.6635	0.8262	0.1709	0.8874
SVM-7	0.1	Linear	0.01	25	0.8251	0.6503	0.8213	0.1683	0.8936
SVM-8	0.1	Poly (2)	0.01	25	0.8185	0.6371	0.8157	0.1687	0.8937
SVM-9	0.1	Poly (3)	0.01	25	0.8015	0.6030	0.7949	0.1754	0.8827
SVM-10	0.1	Linear	0.1	25	0.7958	0.5917	0.7931	0.1760	0.8712
SVM-11	0.1	Poly (2)	0.1	25	0.8346	0.6692	0.8299	0.1580	0.9037
SVM-12	0.1	Poly (3)	0.1	25	0.8147	0.6295	0.8078	0.1707	0.8905
SVM-13	0.1	Linear	0.1	10	0.7958	0.5917	0.7931	0.1760	0.8712
SVM-14	0.1	Poly (2)	0.1	10	0.8185	0.6371	0.8157	0.1687	0.8937
SVM-15	0.1	Poly (3)	0.1	10	0.8147	0.6295	0.8078	0.1707	0.8905
SVM-16	0.01	Poly (2)	0.1	25	0.8346	0.6692	0.8299	0.1580	0.9037
SVM-17	1	Poly (2)	0.1	25	0.8204	0.6408	0.8173	0.1817	0.8797
SVM-18	0.1	Poly (3)	0.1	25	0.8147	0.6295	0.8078	0.1707	0.8905
SVM-19	0.1	Poly (2)	1	25	0.8280	0.6616	0.8240	0.1564	0.9029
SVM-20	0.1	Poly (2)	0.01	25	0.8308	0.6616	0.8250	0.1638	0.8972
SVM-21	0.1	Poly (2)	0.1	100	0.8318	0.6635	0.8262	0.1709	0.8874
#1 SVM-22	0.1	Poly (2)	0.1	5	0.8346	0.6692	0.8299	0.1580	0.9037
SVM-23	0.01	Poly (2)	0.1	100	0.8223	0.6446	0.8199	0.1589	0.8956
SVM-24	10	Poly (2)	0.1	100	0.7958	0.5917	0.7931	0.1945	0.8564

13.7. Assessment 2 – Work Distribution Table

Team Member	Completed Sections	Extra Contributions
Manjyot Joher - 12897981 ***(Team lead)***	• Executive Summary • 7.1.1 – Stage One: Business Understanding & Problem Analysis • 7.1.4 – Stage Four: Model Training • 7.1.5 – Stage Five: Comparative model evaluation • 8.2.2/3 – Backward/Stepwise Logistic Regression • 13.2.3/4 – BLR/SLR config tables • 8.5 – Random Forest • 13.5 – Random Forest • 9 – Comparative model assessment • Conclusion	• Document collation, editing and formatting (25 pages) • In-text citations and referencing • 7.1.2 – Stage Two: Data Exploration
Vi Nguyen - 13592629	• 8.1 – Neural Network • 13.1 – Neural Network • 8.6 – Support Vector Machine • 13.6 – Support Vector Machine • 12 – Challenges encountered	• 9 – Comparative model assessment

	• 7.1.1 – Stage One: Business Understanding & Problem Analysis • 7.1.3 – Stage Three: Data preparation	
Aiden Ye Yint Hlyan – 14017432	• 8.3 – Decision Tree • 13.3 – Decision Tree • 7.1.2 – Stage Two: Data Exploration • 9 – Comparative model assessment • 12 – Challenges encountered	
Su Myat Than Cin - 13486927	• 8.2.1 – Forward Logistic Regression • 13.2.1 – LR parameter descriptions • 13.2.2 – FLR config tables • 9 – Comparative model assessment • 12 – Challenges encountered	• 8.4 – Gradient Boosted Tree • 13.4 – Gradient Boosted Tree
Ye Min Oo – 13506858	• 10 – Model deployment • 7.1.6 - Stage Six: Model deployment • 8.4 – Gradient Boosted Tree • 13.4 – Gradient Boosted Tree • 12 – Challenges encountered	• 9 – Comparative model assessment • In-text citations and referencing

14. References

In-text citations and referencing present within this report utilise APA 7th reference styling.

Arora, S. (2022, July 26). *Let's solve Overfitting! Quick guide to cost complexity pruning of decision trees*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/10/cost-complexity-pruning-decision-trees/

Bhalla, D. (n.d.). *Difference between CHAID and CART*. ListenData. https://www.listendata.com/2015/03/difference-between-chaid-and-cart.html

Campos, M. M., & Pereira, M. C. (2008). Impact of the recent reform of the Portuguese public employees' pension system. Research Papers in Economics. https://www.bportugal.pt/sites/default/files/anexos/papers/wp200812.pdf

Cross Validated. (n.d.). *How is Kolmogorov Smirnov a good measure of classifier performance?* https://stats.stackexchange.com/questions/588928/how-is-kolmogorov-smirnov-a-good-measure-of-classifier-performance

Cutoff node. (n.d.). SAS Help Center. https://documentation.sas.com/doc/en/emref/14.3/n1qmjdusj37md5n1as50qvl0tram.htm

Dang, X. T., Hirose, O., Saethang, T., Tran, V. A., Nguyen, L. A., Le, T. K., Kubo, M., Yamada, Y., & Satou, K. (2013). A novel over-sampling method and its application to miRNA prediction. *Journal of Biomedical Science and Engineering*, *06*(02), 236-248. https://doi.org/10.4236/jbise.2013.62a029

Dataiku. (2023, September 28). *Decision tree interpretation*. Dataiku Community. https://community.dataiku.com/t5/Using-Dataiku/Decision-Tree-Interpretation/td-p/14878

Devansh. (2021, August 25). *Bayesian Information Criterion/Schwarz Criterion Introduced. Machine Learning Terms Episode 1* [Video]. YouTube. https://www.youtube.com/watch?v=kmIJIYF8C_E

Frost, J. (2022). Kurtosis: Definition, leptokurtic & platykurtic. Statistics By Jim. https://statisticsbyjim.com/basics/kurtosis/

Glen, S. (n.d.). Kurtosis: Definition, leptokurtic, platykurtic. Statistics How To. https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/kurtosis-leptokurtic-platykurtic/

Google for Developers. (2022, July 18). *Classification: Accuracy*. https://developers.google.com/machine-learning/crash-course/classification/accuracy

Hendricks, R. (n.d.). *What is a good accuracy score in Machine Learning?* deepchecks. https://deepchecks.com/question/what-is-a-good-accuracy-score-in-machine-learning/

Herbsthofer, L. (2021, January 8). *Reason of having high AUC and low accuracy in a balanced dataset*. Stack Overflow. https://stackoverflow.com/questions/38387913/reason-of-having-high-auc-and-low-accuracy-in-a-balanced-dataset#:~:text=The%20ROC%20curve%20is%20biased,low%20number%20of%20true%20negatives

K2 Analytics. (2020, August 15). *Concordance, Gini coefficient and goodness of fit*. https://www.k2analytics.co.in/concordance-gini-and-goodness-of-fit/#:~:text=In%20machine%20learning%2C%20the%20Gini,the%20better%20is%20the%20model

Kenton, W. (2023, April 12). Kurtosis definition, types, and importance. Investopedia. Retrieved September 7, 2023, from https://www.investopedia.com/terms/k/kurtosis.asp

KHAL, Y. E. (2021, March 18). *Confusion matrix, AUC and ROC curve and Gini clearly explained*. Medium. https://yassineelkhal.medium.com/confusion-matrix-auc-and-roc-curve-and-gini-clearly-explained-221788618eb2#:~:text=AUC%20is%20the%20area%20under,model%20has%20a%20negative%20sign

Khan, R. (2017, December 16). *RPubs - Binary classifier evaluation metrics: Error rate, KS statistic, AUROC, lift, gains table*. https://rpubs.com/riazakhan94/ksroclift

Klima, K. (2021, March 21). Normality testing - Skewness and kurtosis. Connect, Learn, and Share | The GoodData Community. Retrieved September 7, 2023, from https://community.gooddata.com/metrics-and-maql-kb-articles-43/normality-testing-skewness-and-kurtosis-241

Kumar, A. N. (2021, January 27). *Why linear regression is not suitable for classification?* Medium. https://medium.com/analytics-vidhya/why-linear-regression-is-not-suitable-for-classification-cd724dd61cb8

Kumari, K. (2022, August 5). *Implement of decision tree using Chi_Square automatic interaction detection*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/implement-of-decision-tree-using-chaid/

LinkedIn. (2023, August 31). What are the advantages and disadvantages of equal-width and equal-frequency Binning methods? Retrieved September 10, 2023, from https://www.linkedin.com/advice/1/what-advantages-disadvantages-equal-width#:~:text=Both%20are%20unsupervised%20binning%20methods,the%20data%20is%20uniformly%20distributed

MLNerds. (2021, June 21). *What is AUC : Area under the curve?* Machine Learning Interviews. https://machinelearninginterview.com/topics/machine-learning/what-is-auc-area-under-the-curve/

Moro, S., Cortez, P., & Laureano, R. (2011, October). Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology [Paper presentation]. European Simulation and Modelling Conference. https://www.researchgate.net/publication/236231158_Using_Data_Mining_for_Bank_Direct_Marketing_An_Application_of_the_CRISP-DM_Methodology

Natekin, A., & Knoll, A. (2013, November). *Gradient Boosting Machines, A Tutorial*. ResearchGate. https://www.researchgate.net/publication/259653472_Gradient_Boosting_Machines_A_Tutorial

Overview of Gradient Boosting. (n.d.). *SAS Help Center*. SAS Help Center. https://documentation.sas.com/doc/en/vdmmlcdc/8.5/vdmmlref/n0qhcrcz16jtk0n1xl3luzanfndx.htm

Parashar, N. (2023, January 11). *What is an accuracy score and how to check it?* Medium. https://medium.com/@niitwork0921/what-is-an-accuracy-score-and-how-to-check-it-13b23eeed6a3

PennState Eberly College of Science. (n.d.). 11.4 - Interpretation of the principal components. Retrieved September 11, 2023, from https://online.stat.psu.edu/stat505/lesson/11/11.4#:~:text=Interpretation%20of%20the%20principal%20components%20is%20based%20on%20finding%20which,of%20course%20a%20subjective%20decision

Prabhakaran, S. (n.d.). *Evaluation metrics for classification models – How to measure performance of machine learning models?* Machine Learning Plus. https://www.machinelearningplus.com/machine-learning/evaluation-metrics-classification-models-r/

Roberts, A. (2023, March 22). *Data Binning challenges in production: How to bin to win*. Arize AI. https://arize.com/blog-course/data-binning-production/

SAS Help Center. (n.d.). *Forest Properties*. https://documentation.sas.com/doc/en/vdmmlcdc/v_017/vdmmlref/n00158lshcvbi9n1h9xc3grkefwm.htm

SAS. (2015, March 13). Descriptive statistics — The more, the merrier in SAS visual analytics. SAS Blogs. https://blogs.sas.com/content/sgf/2015/03/13/descriptive-statistics-the-more-the-merrier-in-sas-visual-analytics/

SAS. (2018, June 25). Interactive binning node vs. transformation node binning. SAS Communities. Retrieved September 10, 2023, from https://communities.sas.com/t5/SAS-Data-Science/Interactive-Binning-Node-vs-Transformation-Node-Binning/td-p/393634

SAS. (2021, April 29). Transformations node in SAS model studio on SAS viya. SAS Communities. Retrieved September 10, 2023, from https://communities.sas.com/t5/SAS-Communities-Library/Transformations-Node-in-SAS-Model-Studio-on-SAS-Viya/ta-p/737959

SAS. (2021, March 4). SAS Viya: Optimal binning. SAS Communities. Retrieved September 11, 2023, from https://communities.sas.com/t5/New-SAS-User/SAS-Viya-Optimal-Binning/td-p/539999

SAS. (n.d.). *Logistic Regression Properties*. SAS Help Center. https://documentation.sas.com/doc/en/vdmmlcdc/v_015/vdmmlref/p0zwag4h3j7hqln1jx5csl5wtwb3.htm

Search.r-project.org. (n.d.). Intracluster correlation coefficients for clustered data. https://search.r-project.org/CRAN/refmans/fishmethods/html/clus.rho.html

Seif, G. (2019, May 21). *Understanding the 3 most common loss functions for machine learning regression*. Medium. https://towardsdatascience.com/understanding-the-3-most-common-loss-functions-for-machine-learning-regression-23e0ef3e14d3

Singh, M. (2021, January 13). *CHAID decision tree*. Medium. https://mksingh0892.medium.com/chaid-decision-tree-30a4c7ba6efc

StatQuest. (2019, July 12). *ROC and AUC, Clearly Explained!* [Video]. YouTube. https://www.youtube.com/watch?v=4jRBRDbJemM

Sturgis, P. (2004). Analysing complex survey data: Clustering, stratification and weights. Social Research Update. Retrieved September 12, 2023, from https://sru.soc.surrey.ac.uk/SRU43.html

TheGlobalEconomy.com. (2023, June). Portugal bank deposit interest rate, percent, June, 2023 - Data, chart. Retrieved September 8, 2023, from https://www.theglobaleconomy.com/Portugal/deposit_interest_rate/

Trevisan, V. (2022, February 28). *Evaluating classification models with Kolmogorov-Smirnov (KS) test*. Medium. https://towardsdatascience.com/evaluating-classification-models-with-kolmogorov-smirnov-ks-test-e211025f5573

UCI Machine Learning Repository. (2012, February 13). Bank Marketing. https://archive.ics.uci.edu/dataset/222/bank+marketing

Walker, C., & Rogers, W. (2020). Principal component analysis demystified. SAS Customer Support Site | SAS Support. https://support.sas.com/resources/papers/proceedings20/5110-2020.pdf

Zach. (2021, September 8). *F1 score vs. accuracy: Which should you use?* Statology. https://www.statology.org/f1-score-vs-accuracy/

Zach. (2022, May 19). *What is a "Good" Accuracy for Machine Learning Models?* Statology. https://www.statology.org/good-accuracy-machine-learning/