QBUS3820 Machine Learning and Data Mining for Business

(2022 Semester 1)

Group Assignment

Due date: Friday 27 May 2022

Group Number: 03

Group Members:

_____500449260_____

_____500080960 _____

_____490449275_____

Table of Contents

Problem formulation

For this project, we consider the classification datasets, both of which revolve around the effectiveness of marketing campaigns.

The first dataset comes from a fashion store. Each row contains information on a customer, and the response is whether said customer responded to a promotional email. The second dataset comes from a bank. Each row corresponds to a phone campaign made to a customer; the response here, is whether the customer subscribed to a term deposit with the bank. These datasets are described in detail below.

For both contexts, a pressing problem is determining in advance, whether a customer is likely to respond to a campaign. This is needed to help efficiently target advertising efforts (for example, advertising costs are poorly used if advertising is made towards unreceptive customers).

This is where machine learning models can be of assistance: by using features from customers (and economic indicators, as in the bank dataset), these models can help predict the responsiveness of a customer. These predictions can be used to assist decision-making by management. For example, if a potential customer is predicted to have a high probability of responding to the email, then management may focus more resources to attract sales from the customer. Likewise, if a customer is predicted as likely to subscribe to a term deposit, the bank could increase campaign efforts to secure the customer.

Therefore, a goal of the project is:

1. **To develop machine learning models that can predict the responsiveness of customers to marketing campaigns.**

   For each dataset, a range of different models are built. Hyperparameters for each model are tuned, using the cross-validation F1 score (see below). To select the final model(s) for each dataset, we evaluate performance on the validation set using a range of metrics. We also consider factors such as interpretability and ease of maintenance. Finally, we evaluate our choice by calculating performance on the test set.

Apart from developing models with high predictive accuracy, it is also important to be able to understand why the model made a certain prediction. This is an ethical requirement in several business contexts: for example, banks utilising modelling for granting credit loans, may be required to explain why a client was denied a loan.

Related to this, our client may also want to have a deeper understanding about the factors potentially influencing customer responsiveness to campaigns. These insights may then be used for guiding marketing efforts. Therefore, another task is

2. **To uncover insights about which factors potentially impact the responsiveness of customers to marketing campaigns**

   For the fashion store dataset, we perform clustering of customers based on spending patterns, and provide evidence suggesting that purchasing patterns are associated with receptiveness.

   For the bank dataset, we discuss the impact of demographics such as job and marital status, on customer receptiveness. We also explore economic factors such as employment and the consumer price index (CPI), showing that there are also linked to receptiveness.

STORE DATASET

Data Preparation

Dataset

This dataset is based on the 'Fashion store' dataset. There are 21,740 rows, each corresponding to a customer characteristic and spending pattern with the store. The response variable is whether the client responded to the promotion, and this will be the classification goal.

There are 20 attributes:

- Discrete: number of purchase visits, number of different product classes purchased, etc.
- Continuous: lifetime average between visits, amounts spent etc.
- Binary: valid phone number (yes/no), credit card user (yes/no), etc.

The data as-is, is not yet ready for learning. Below, we address several issues prior to modelling. Note that we use **bold type** for **names of variables**.

Data Pre-processing

Upon inspection, there are no missing values. **VALPHON**, a categorical variable with two categories 'yes' and 'no', was converted to value 1 for people with a valid phone number and 0 otherwise.

Type conversion

There are continuous, discrete, and binary variables in our dataset. Variables belonging to each datatype are summarized in table S.1 (see appendix). Specifically, the data set consists of 32 continuous, 11 discrete and 4 binary variables including the response variable, which is **RESP**. With so many variables, we take the approach of variable filtering, as our attempt to exclude irrelevant/redundant variable from our models.

Outliers

According to box plots in figure S.1 and figure S.2 (see appendix), most continuous variables are right skewed with large numbers of outliers. It was decided to retain outliers as they may contain useful information. Furthermore, outliers are not a problem for the tree-based methods; for the linear model feature engineering such as log transformation can be utilized to mitigate the issue.

Exploratory Data Analysis

Response Variable

As mentioned in the problem formulation, **RESP** indicates whether a customer responded to the marketing campaign of the store. There is class imbalance, since only with 16.6 % of customers in the survey responded to the marketing campaign.

Continuous & Discrete Variables

Most numerical variables are right skewed as shown in figure S.3. (see appendix). However, **GMP** is left-skewed with a skewness of -1.354. From earlier, we mentioned that most numerical variables have outliers to the right. Therefore, it is reasonable to expect most of variables will have high kurtosis. Indeed, there are

many numerical variables with kurtosis of at least 3 and in the outstanding case, **PCOLLSPND** has kurtosis of 9.777 as shown in table S.2 (see appendix).

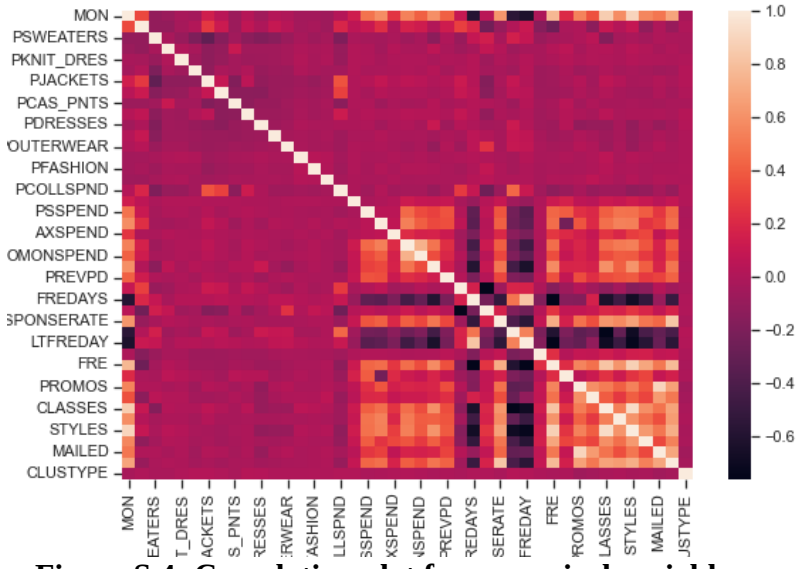


Figure S.4: Correlation plot for numerical variables

From figure S.4, several numerical variables have high correlation. For instance, **LTFREDAY** and **MON** have high negative correlation of -0.630. Therefore, this may violate the 'no multicollinearity assumption' made in linear models such as the logistic regression. To mitigate this, we adopt the penalized logistic regression below.

By observing the sigmoid plot in figure S.5 (see appendix), several continuous variables are suggested t be strongly associated with the response. For example, the sigmoid curve for **LTFREDAY**, has a steep negative slope – indicating a small change drastically reduces the log-odds of a response. We can interpret the slope as saying, as the time between shopping days increase, the chance of customer being responsive is expected to decrease. In contrast, **MON** and **SMONSPEND** are positively associated with the log-odds of response (positive slope). This is reasonable, since total net sales and the amount spent in the past six month should be associated with customers responsiveness.

Similar analysis was conducted for discrete variables (figure S.6, see appendix). For instance, the variables **FRE, STYLES, CLASSES** and **RESPONDED** are suggested to be important for modelling given their steep sigmoid functions.

Table S.4 illustrates the top 4 continuous and discrete variables with the highest mutual information (see table S.5 in appendix for other variables).

Continuous variable	Mutual information	Discrete Variable	Mutual information
LTFREDAY	0.110	FRE	0.081
FREDAYS	0.064	STYLES	0.076
MON	0.053	CLASSES	0.062
SMONSPEND	0.050	RESPONDED	0.052

Table S.4: top 4 continuous variables and discrete variables with the highest mutual information

As observed, mutual information aligns with the sigmoid plots, in the sense that variables with a high mutual information were also noted to have steep sigmoid functions. However, **RESPONDED**, despite having integer values has too few distinct values to be considered numerically (figure S.18 in appendix). Further steps are taken in feature engineering section.

Binary Variables

Table S.6 summarizes the top three binary variables with the highest mutual information. This agrees with the crosstab plots, where **WEB** and **VALPHON** did not seem useful for predicting the response (Figure S.17).

BINARY VARIABLES	MUTUAL INFORMATION
CC_CARD	0.028
WEB	0.009
VALPHON	0.006

Table S.6: mutual information for binary variables

Feature Engineering

New Feature Generation

RESPONDED_6

As mentioned, it is reasonable to treat **RESPONDED** as a categorical variable due to its low number of distinct values. All values equal to or above 6 were be merged due to their low frequencies. The variable is was then one-hot encoded.

As **RESPONDED_6** has the highest metric as shown in table S.7 (see appendix), it is suggested to be useful for modelling

TOTAL_SPENDING

Intuitively, spending in different stores as measured by **AMSPEND, PSSPEND, CCSPEND, AXSPEND** may not be relevant to **RESP** due to customer's arbitrary preference for each store (for example, a customer may spend the most at the AM store given he/she lives close to it). Hence, it is motivated to come up with a new variable named **TOTAL_SPENDING**, which is the total amount customers spent in all stores. This feature has higher mutual information than the amounts spent at individual stores (see appendix for full list).

VARIABLE	MUTUAL INFORMATION
TOTAL_SPENDING	0.056

Table S.8: mutual information for TOTAL_SPENDING

SPENDING_PLEGWEAR & SPENDING_PSWEATERS

The fraction spent on a particular product may not be as useful as the actual amount spent on that product. Therefore, it motivates the formations of interaction terms between **MON**- total net sales and other spending fraction variables such as **PSWEATERS, PKNIT_TOP, and PKNIT_DRES**. Nevertheless, including all interaction terms as official inputs is not an ideal approach since it may lead to the curse of dimensionality and multicollinearity due to the same underlying **MON** factor. Therefore, the report only adopts two interactions terms with the highest mutual information as shown in table S.9 (see appendix).

Data Transforming

Log Transformation

As most of the chosen numerical variables have right-skewed distributions, it is reasonable to apply logarithm transformation. Furthermore, some variables such as **MON** may have high variation and logarithm transformation turns out to be the appropriate tool for reducing such variable and magnitude, which helps improve the learning process.

Min-Max Scaling

In this report, in the next section, deep feedforward neural network and L_2-Penalized logistic regression are adopted. This indicates the need for standardised data. Nevertheless, instead of using standardisation, the Min-Max scaling is utilized. Reasonably, Min-Max scaling may prevent the supervised learning models from getting biased toward a specific range of values by ensuring that training data points fall into zero to one interval. For example, if the trained model is based on logistic regression and features are not scaled, some features having higher impacts than others would affect the prediction performance by giving undue advantage for some variables. Consequently, this puts certain classes at a disadvantage while training model.

Feature Screening

This section will give the readers a summary of which inputs will be used to build the models. As mentioned in the exploratory data analysis section, features found potentially useful for the modelling are **LTFREDAY, FREDAYS, MON, SMONSPEND, FRE, STYLES, CLASSES**, and **CC_CARD**. The engineered features **TOTAL_SPENDING, SPENDING_PLEGWEAR, SPENDING_PSWEATERS, and RESPONDED_6** were also suggested to be important, as shown.

Loss Matrix

There is an imbalance between the positive class and the negative class in the output variable. Hence, most machine learning models would be biased toward the larger class (i.e., negative class). To resolve, it is necessary to penalize the bias of the models by setting the loss of false negative larger than the loss from false positive. With simple calculation, we can come up with the ratio between the number of responsive customers to the number of nonresponsive customers is approximately 0.199. Without loss of generality, we can assume that the loss of false positive is 1 unit, which makes it reasonable to approximate the loss of false negative is 5.024, the inverse of 0.199.

ACTUAL\PREDICTED CLASS	POSITIVE	NEGATIVE
POSITIVE	**0**	**5.024**
NEGATIVE	**1**	**0**

Table S.10: loss matrix

Model Building

Six statistical machine learning models are adopted, which are logistic regression as a baseline model, L_2-penalized logistic regression, light gradient boosting machine, random forest, a stack model of three previous models, and deep feedforward neural network. Further details would be discussed in each subsection.

The main performance metric used for hyperparameter tuning will be the F1 score. As mentioned, there is class imbalance in the dataset, so our measure of performance should take this into account. The F1 score

achieves a balance between precision and recall, hence we saw it as more appropriate for this task than metrics such as AUC and accuracy (which do not take into account the differences in distribution).

L_2-Penalized Logistic Regression

Rather than the traditional logistic regression, the L_2 penalized logistic regression is used to mitigate overfitting with the presence of the regularization term.

Regularization also helps mitigate the problem of multicollinearity, which we noted. Specifically, multicollinearity may cause problems such as inflated coefficient estimates, so the L2 penalty solves this problem by restricting the estimation of coefficients to a sphere (James, G. et al., 2021)

In this model, there are two hyperparameters that need tuning, one of which is the penalty hyperparameter λ. The grid search method is applied here with 40 values attempted in the range $[0.1, 30]$ and the optimal value for such hyperparameter is 0.1, which has the highest mean F1 score and lowest standard deviation as shown in table S.16 (see appendix). Next, the threshold boundary is the remaining concerned hyperparameter. This is because the default figure 0.5 may not work well in situations with class imbalance. Here, the grid search method is used along with cross validation. Specifically, the report creates a list of 100 equally spaced possible values ranging from 0.2 to 0.7. For each number, cross validation is performed to obtain the F_1 score for each iteration. The average F_1 score for each number is the basis for comparison such that the higher the metric, the better the threshold. It turns out 0.266 is the optimal boundary with the score of 0.529 as shown in figure S.7 (see appendix).

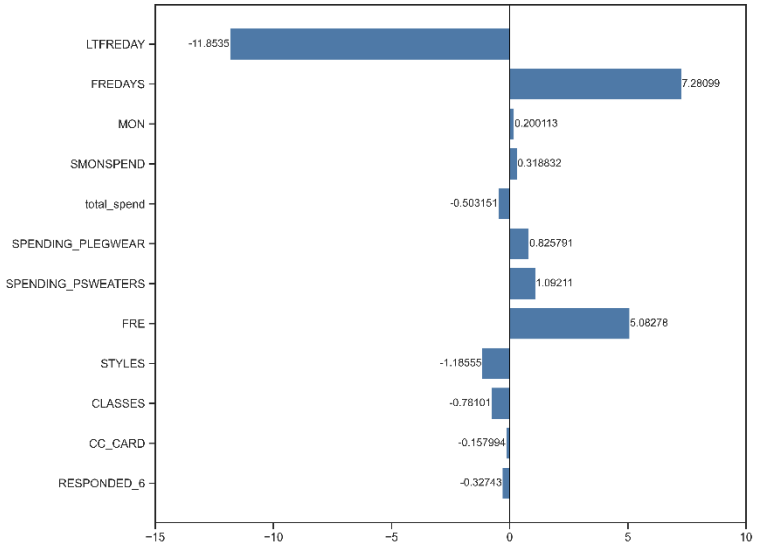


Figure S.8: Coefficient plot

Next, the L_2 penalized logistic regression model is fitted on the whole training data set. Briefly speaking, the horizontal bar plot in figure S.8 illustrates the summary of the coefficients' values of the inputs. As expected in the exploratory data analysis section, the high positive coefficients of **FREDAYS**, and **FRE** shows that the higher the values of these variables are, the greater the chance of response. In contrast, the huge negative coefficient of **LTFREDAY** suggests the opposite direction.

Besides, the effect of new features in the feature engineering section is also consolidated by this graph where the coefficients of the engineered features match the motivations for such creations. For instance, the positive sign for the coefficients of two numerical variables **SPENDING_PLEGWEAR** and **SPENDING_PSWEATER** aligns with the previous observation that the spending amount rather than spending fraction will be more useful in predicting the response.

Light Gradient Boosting Machine

Although the penalized logistic regression has high interpretability, by default, it does not consider potential interaction effects. Thus, light gradient boosting machine or LightGBM, which is a variation of gradient boosting method, is adopted as it accounts for possible interactions among inputs.

LightGBM consists of many hyperparameters that need to be tuned simultaneously. For example, the number of decision trees, maximum number of leaves for each tree, minimum number of examples in each leaf, size of the penalty terms, subsample rate, etc.

Facing such computational challenge, the report adopts Bayes optimization. Table S.11 illustrates the ranges considered for each hyperparameter, over which the algorithm optimizes the cross-validation F_1 score. As indicated in the table S.11, the report restricts the number of trees to less than or equal to 1000 since increasing number of trees in LightGBM can cause overfitting, which is different from Random Forest where large number of trees may not bring such disadvantage and since trees in LightGBM are slow to overfit then to avoid underfitting, the number of trees can be at least 500. Other than that, as one desirable goal is to reduce the variance of predictions in each leaf, it is necessary to force the minimum number of examples in each leaf to be at least 30 and at most 50.

HYPERPARAMETER	TYPE	POSSIBLE VALUES	TUNED VALUE
Number of trees	Integer	[500,1000]	550
Number of leaves	Integer	[5, 40]	6
Minimum number of examples	Integer	[30, 50]	30
Alpha	Float	(0,0.01]	6.765×10^{-4}
Lambda	Float	(0,0.0001]	1.121×10^{-8}
Subsample	Float	[0.4,1]	0.900
Column sample	Float	[0.8,1]	0.821
Positive weight	Float	[1,10]	2.578

Table S.11: Summary of Bayes optimization's input and output for LightBGM's hyperparameters

There is another point worth considering. In the Bayes optimization algorithm, we can tune the threshold for prediction using a hyperparameter called "class_weight". This indicates the relative importance of class 1 over class 0. For instance, if "class_weight = {0:1, 1:4}", then class 1 is 4 times more important than class 0, which makes the loss of false negative is 4 times higher than the loss of false positive and this leads to the value of $\tau = \frac{1}{1+4} = 0.2$. The report does not tune "class_weight" directly but instead fixes the weight of class 0 equal to 1 and tunes the second argument, which is the weight of class 1 and called ***positive weight*** hyperparameter in this report.

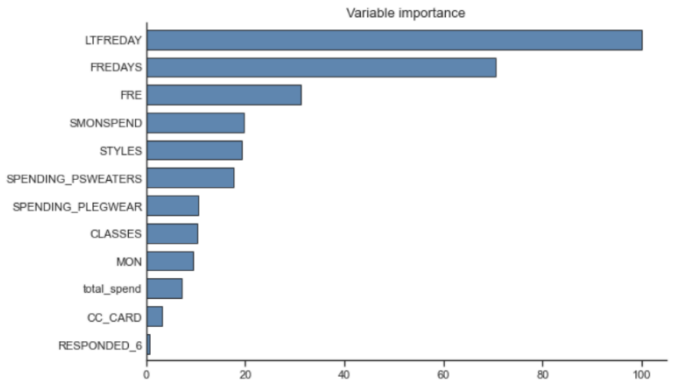


Figure S.9: Feature importance plot for LightGBM

After hyperparameter tuning, the LightGBM model is fitted on the whole training data set. Figure S.9 shows most important variables, ranked by variable importance – a larger variable importance corresponds to a variable which is more important for model predictions.

There are commonalities between the important variables here, and those deemed important by the logistic regression (based on absolute value of their coefficients). For example, **FRE** has the 3rd largest coefficient in terms of magnitude and is also ranked at 3rd place by variable importance in the LightGBM.

Random Forest

As mentioned in LightGBM section, unnecessarily high number of trees in LightBGM can make the model suffer from overfitting due to the dependence between the current tree and the previous tree. As it is the case, random forest model will be a potential model to fix this problem since each tree is constructed independently from other trees.

Random forest also has various types of hyperparameter. For example, the objective function must be determined, which impacts the choice of splitting variables and splitting points. In this report, only two objective functions are considered, which are entropy and Gini-index due to their robustness in measuring purity of the leaves.

To prevent high variations in predictions, the report limits the minimum number of examples in each leaf to be between 30 and 50. To prevent any decision tree overfitting or underfitting, it is necessary to narrow the maximum number of leaf nodes in the interval between 10 and 20.

Based on optimization results in LightGBM section and also the loss matrix, we can also save computational cost for optimizing the positive class weight, by focusing the interval between 2 and 5. On the other hand, the number of trees will be set to 100 to boost the learning process and then the optimal model will be fitted with 2000 trees. The summary of those restrictions in hyperparameters is shown in table S.12 (see appendix)

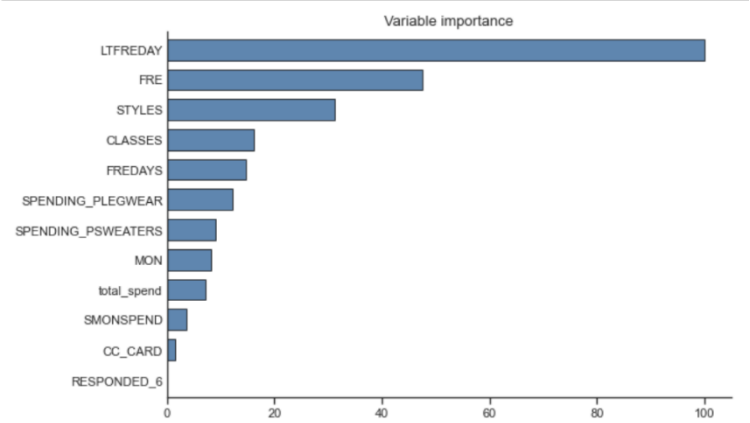


Figure S.10: Feature importance plot for random forest

Figure S.10 indicates the resultant variable importances from the Random Forest. Similar to the Penalised Logistic regression and LightGBM, **LTFREDAY** and **FRE** still play the most critical roles here.

Model Stacking

As LightGBM and random forest are representatives for boosting and bagging methods respectively, the family of model ensembles methods also has model stacking, which theoretically can combine the strength and neutralize the weakness by considering outputs of individual classifiers as the inputs of a higher-level classifier. Therefore, model stacking will be considered.

To maintain accuracy of the stack model, LightGBM and random forest with their previously tuned hyperparameters will be selected as individual classifiers; the L_2-penalized logistic regression model will play the role as the meta model or the 'combiner' to increase the interpretability of the model. Figure S.11 summarises the model stacking fitting procedure.

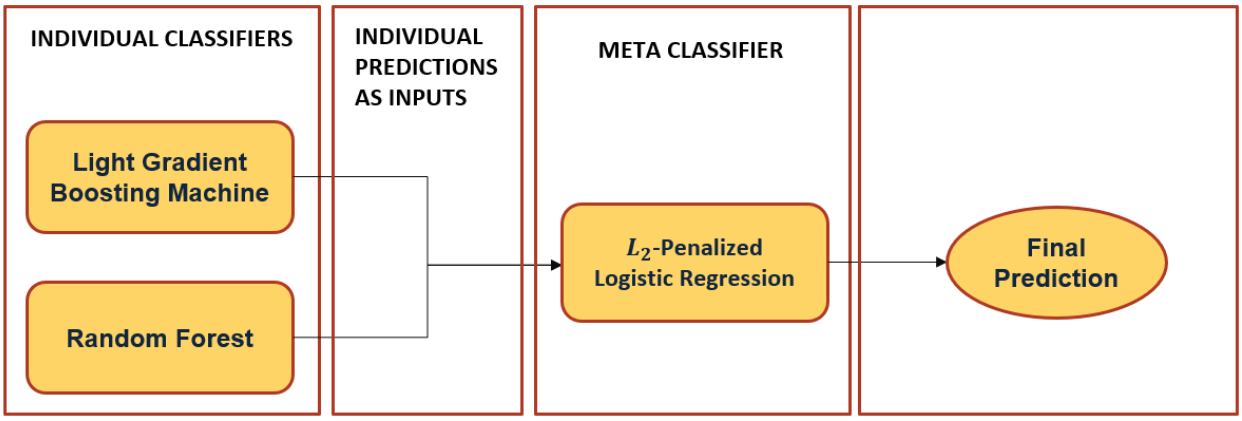


Figure S.11: the stacking process

Deep Feedforward Neural Network

A deep feedforward neural network will be explored in this section, as these are known to have high predictive accuracy. Also, in many models, one must rely significantly on feature engineering,

which is not always successful without adequate domain knowledge. As a result, the report is inspired to use a representation learning approach: the deep feedforward neural network can flexibly learn new features by using its hidden layers.

Technically, it has been shown that deep networks have an exponential advantage over single layer networks for certain cases. Therefore, a neural network with 9 hidden layers with 128 units in each layer will be built. In terms of activation function, instead of using tanh and sigmoid, which may suffer from vanishing gradient issue during the fitting process, or ReLU function, which still have the drawback that gradient-based learning gets no information from training examples with zero activation, ELU function with $\alpha = 1$ will be adopted here to mitigate problems and ensure that the activation function is smooth, which helps speed up the learning process (see appendix to see the explicit form of the ELU activation functions. With respect to the training process, 100 epochs will be utilized to make sure the estimated parameters are reasonably good in minimizing the loss function.

Model Selection

To select the best model(s) out of those that we have built, we compare model performances on the validation set. We also compare qualitative aspects such as interpretability and ease of maintenance, as it is important to factor in practical considerations for the business context when selecting the best model(s).

Performance on validation set

Table S.13 summarizes performance of our models the validation set, using different metrics (Table S.13). In terms of error rate, surprisingly, the baseline logistic regression model happens to perform the best, with an error rate of 0.148 (Table S.13). Nevertheless, recall that there is class imbalance: this makes any biased model have low error due to its tendency of favoring negative class prediction.

Therefore, metrics which consider the class imbalance should be favored instead. Here, we discuss the AUC, F1 score and Average loss (from the loss matrix).

If we were to select models based on their performance on these three metrics, then the Deep Feedforward Neural network is the optimal choice. This obtains the highest F1 (0.542), best AUC (0.857) and the lowest Average loss (857.544).

On the other hand, the L2-penalised logistic regression also performs decently on these metrics. Despite being simpler than the LightGBM, Model stack and Deep Feedforward Neural Network, it attains a comparable F1 score of 0.533. The AUC and Average Loss are also comparably just as good (0.854 and 932.168, respectively).

The Deep Feedforward Neural network is optimal in terms of validation F1, AUC and Average Loss. However, as we shall see in the next section, the L2-penalised logistic regression is the more favored model, after considering qualitative aspects such as interpretability and ease of maintenance.

Model	Error Rate	Sensitivity	Specificity	Precision	Recall	F1 score	AUC	Average Loss
L_2-Penalized Logistic Regression	0.184	0.634	0.852	0.460	0.634	0.533	0.854	932.168
Random Forest	0.185	0.546	0.869	0.453	0.546	0.495	0.837	1061.936
LightGBM	0.170	0.596	0.876	0.490	0.596	0.538	0.855	957.504
Model Stacking	0.178	0.618	0.863	0.473	0.618	0.536	0.855	941.312
Deep Feedforward NN	0.198	0.706	0.821	0.440	0.706	0.542	0.857	857.544
Logistic Regression – Baseline	0.148	0.277	0.966	0.621	0.277	0.383	0.854	1372.264

Table S.13: model validation

Deployment considerations

The business context also plays an important role. Aspects such as interpretability and maintenance cost are summarized in Table S.14.

Despite high performance on validation metrics (table S.13), the model stack and neural network require considerable resources to train and maintain, since these are relatively complex models. For example, given the abundance of parameters in the neural network, large amount of effort needs to be given to maintain such a model. On the other hand, the L_2 penalized logistic regression model is also the easiest to maintain, being the simplest.

Model	Interpretability	Maintenance cost
L_2-Penalized Logistic Regression	High	Inexpensive
Random Forest	Medium	Expensive
Light Gradient Boosting Machine	Medium	Expensive
CatBoost	Medium	Expensive
Model Stacking	Low	Highly expensive
Deep Feedforward Neural Network	Low	Highly expensive

Table S.14: qualitative factors consideration. CatBoost is considered as well, as we use this in the bank dataset.

Another consideration is interpretation. Among all models, the L_2 penalized logistic regression is the most interpretable, which makes it easy for management to interpret predictions from the model (without requiring much technical knowledge).

As we have shown on the validation set, the L_2 penalized logistic regression model does not perform much worse than its counterparts. Thus, considering interpretability and maintenance cost as well, this makes the L_2 penalized logistic regression the optimal model we recommend.

Model Evaluation

To evaluate the decision made in the previous section, we refit all 6 models using the combined train and validation set. Following this, we measure the same performance metrics on the test set.

Table S.15 summarizes these performance metrics on the test set. Now as it turns out, the L_2 penalized logistic regression looks to be one of the best models in terms of F1, AUC and Average Loss! This model attains the highest F1 score of 0.525, a very high AUC of 0.851, and the second lowest average loss of 932.024. These performances justify our choice of L_2 penalized logistic regression as the optimal model.

	Error Rate	Sensitivity	Specificity	Precision	Recall	F1 Score	AUC	Average Loss
Deep Feedforward Neural Network	0.218	0.695	0.799	0.408	0.695	0.514	0.851	916.640
L_2-Penalized Logistic Regression	0.195	0.651	0.835	0.440	0.651	0.525	0.851	932.024
LightGBM	0.185	0.598	0.858	0.457	0.598	0.518	0.852	985.480
Random Forest	0.189	0.590	0.855	0.448	0.590	0.510	0.842	1005.552
Model Stacking	0.193	0.634	0.842	0.444	0.634	0.522	0.851	950.168
Logistic Regression -Baseline	0.138	0.316	0.971	0.683	0.316	0.432	0.852	1293.928

Table S.15: model evaluation

Data Mining

Responsiveness of customers to campaigns is based on their spending patterns

It is reasonable to believe that customers with different spending habits, may respond differently to marketing campaigns. For example, if the store has an extensive collection of sweaters, then customers interested with sweaters will probably respond more than other groups.

To verify this, we performed a clustering of the customers based on the clothing categories they purchased. This information is recorded in variables such as PSWEATERS, PKNIT_TOPS etc. These measure the proportion a customer spent on a particular clothing category, such as sweaters and knit tops.

For our clustering, we used the Manhattan distance, since this is relatively robust to outliers. We also used Wards linkage, as this is known to be a good default linkage. The cluster heatmap is provided in the Appendix (Figure S.16).

From observation, the most intuitive clustering was to cut the corresponding dendrogram to give three clusters. This clustered customers into three groups: *those which spent a large proportion on jackets, class career pants, and collectible line clothing; those which spent a lot on sweaters, and other customers*. We extracted the cluster labels into a variable **product_cluster.**

We used a Chi-squared test for independence to test for association between these clusters and the count of responded customers. We also ran a one-way ANOVA, to test for difference in the mean response across clusters. Both tests suggest that receptiveness differs across clusters (Table S.16).

TEST	H_0	P-VALUE	DECISION
Chi-Squared	RESP does not depend on product cluster	1.523×10^{-36}	Rejected null hypothesis
One-way ANOVA	Response rates are the same across different clusters	4.090×10^{-104}	Rejected null hypothesis

Table S.16: Chi-Squared test and One-way ANOVA test

With a difference in mean response identified, the next step would be to identify which clusters (if any) have greater response than the others. Results from Tukey's range test are summarized in Table S.17.

GROUP 1	GROUP 2	MEAN DIFFERENCE (1-2)	95% CI
Jackets, pants, and collectibles	other	5.2334	[4.2573, 6.2095]
Jackets, pants, and collectibles	sweater	-4.8812	[-6.2605, -3.502]
other	sweater	-10.1146	[-11.2957, -8.9336]

Table S.17: Turkey's range test for testing difference in mean response between each of the clusters

Interpreting Table S.17, we are (at least) 95% confident that customers purchasing sweaters have a significantly higher response rate than the other two clusters. Further, response rate for customers purchasing jackets, pants and collectibles is also significantly higher than the remaining customers.

Therefore, the store is strongly advised to focus marketing campaigns on customers predominantly purchasing sweaters, or jackets, pants, and collectibles.

BANK DATASET

Data Preparation

Dataset

The bank dataset is based on the 'Bank Marketing' dataset. There are 41,188 rows, each corresponding to a direct marketing campaign (phone call) to a customer. The response is whether the client subscribed to a term deposit, and this will be the classification goal.

There are 20 attributes:

- Discrete: job, marital status, education, etc.
- Continuous: last contact month, last contact duration (in seconds), etc.
- Binary: phone type, housing loan, etc.

An important subtlety of the data is that rows are ordered by date (May 2008 to November 2010), which we will return to below. The data as-is, is not yet ready for learning. Below, we address several issues prior to modelling.

Missing values

From the data dictionary, there are missing values in several categorical attributes, all coded with the "unknown" placeholder. These attributes are **job**, **marital**, **education**, **default**, **housing**, and **loan**. Figure B.1 (see appendix) visualises the missing values.

As mentioned, observations are ordered by date, but missing values do not appear to be systematic with respect to this. Further, observations are collected from distinct clients, so it is reasonable they are relatively independent. This means applying deletion technique would not affect the general pattern.

Given their low percentage of missing values, we removed rows with missing values in **job**, **marital**, **education**, **housing**, and **loan** (Figure B.1). **Default** however, had a substantial percentage of missing values (20.9%). Hence, the report treats these missing values as a level in themselves, encoding them with 'unknown'. As a result, 38,245 observations remained.

Outlier analysis

Like the store dataset, some attributes such as **age** are right skewed with notable number of outliers, evident by the box plots of figure B.2 (see appendix). It is decided those outliers are kept for the same reasons as discussed in the store dataset.

Further Processing

As mentioned in the dictionary, duration (duration of last contact, in seconds) is only known after the client decided to subscribe or not, so it should not be a predictor in a realistic model. Therefore, we dropped this variable.

Since observations are ordered by date, it is important to remove potential time-related effects. Therefore, we randomly permuted the order of observations. This is also important for the CatBoost encoder, which is sensitive to the ordering.

Train/validation/test split

There exists class imbalance in the response since only around 11% of campaigns resulted in a subscription. Hence, we used a stratified split to ensure equal class proportions in the training, validation, and test sets. 70% (n = 26,771) of the data was kept for training. From the remaining observations (n = 11,474) 70% was used as the validation set, with the remaining used as the test set.

Exploratory Data Analysis

Bivariate relationships between continuous variables and subscription

Figure B.3 (see appendix) displays the resulting sigmoid functions from fitting logistic regression models to each of the continuous attributes. At first glance, most tend to have considerable negative associations with the output, especially the social and economic factors (e.g., euribor3m rate, consumer confidence index). On the other hand, bivariate plots between age and output suggests positive relationship, but the pattern is not particularly strong.

Multicollinearity

Figure B.4 (see appendix) displays the correlation between the continuous variables. Most variables have a relatively low correlation (below r = 0.5). However, among euribor3m, emp.var.rate, cons.price.index, and `nr.employed` there are several very strong correlations larger than r = 0.9 with the strongest figure observed between euribor.3m and emp.var.rate at 0.972.

Multicollinearity is not a problem for the tree-based methods. This is because the model in nature follows the greedy principle and only considers the split that would minimise error. It is, however, a problem for linear models, including logistic regression. We attempt to mitigate the issue by adopting the regularised logistic regression below.

High cardinality and sparse levels

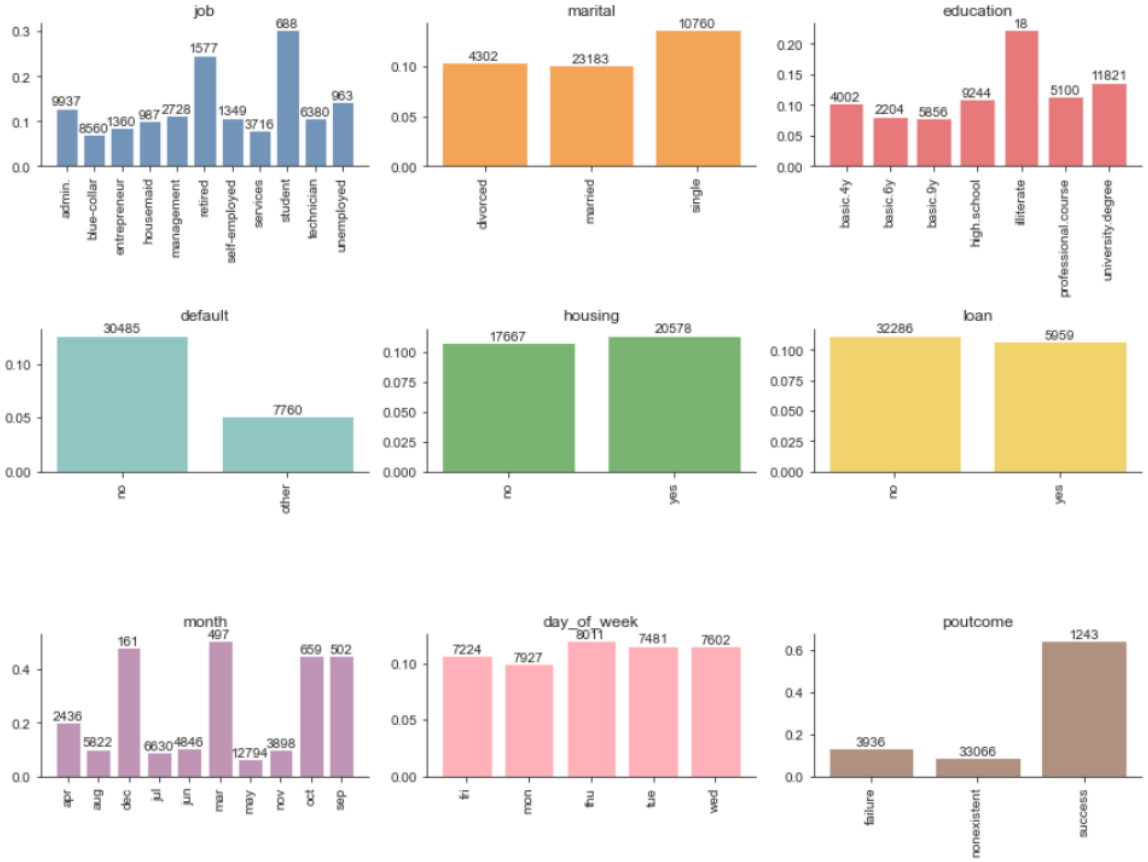


Figure B.5: crosstab plots showing the mean response for levels of each categorical variable.

Figure B.5 shows the subscription rate grouped by levels of each of the categorical variables. Some categorical variables have very different subscription rates among their levels, suggesting these variables are useful for predicting subscription. These include job, marital, education, default, month, poutcome, and contact. On the other hand, housing and loan seem to be irrelevant towards predicting subscription, since retention rates appear similar for their levels. However, further consideration for quantitative mutual information is needed for final claim.

Furthermore, a challenge is high cardinality (many levels), for example as exhibited in job, education, month and day of week. Another challenge is the sparse levels (levels with few observations). For example, only three observations have default = yes (Figure B.5). These issues are handled in the feature engineering section.

Mutual Information

Table B.6 summarises the variables with the ten highest and two smallest mutual information with respect to term deposit subscription. The social and economic-related variables all rank the highest in terms of mutual information; these seem to have higher association with subscription than any of the other continuous or categorical variables.

On the other hand, among categorical variables, poutcome, month and contact have the highest mutual information– this reinforces suggestions from figure B.5 that poutcome, month and contact are useful for

predicting subscription. Finally, the table emphasises the weak relationship of housing and loan with the output, hence they are no longer considered subsequently.

Variable	Type	Mutual information
euribor3m	continuous	0.071
cons.conf.idx	continuous	0.066
cons.price.idx	continuous	0.066
nr.employed	continuous	0.063
emp.var.rate	continuous	0.056
pdays	continuous	0.037
poutcome	categorical	0.029
Month	categorical	0.026
contact	categorical	0.011
…		
housing	categorical	0.000
loan	categorical	0.000

Table B.6 Mutual Information

Feature engineering

New levels of default

As mentioned in the EDA, only three observations have defaulted (Figure B.5). This is too small a sample to do any sort of inference on by its own. We shall pool these observations with those that have default 'unknown'.

CatBoost Encoding

As mentioned, high cardinality is present in some variables, notably education, job, month, and day of week, which prove detrimental to tree-based models. To overcome the high cardinality, we utilise CatBoost encoding. Levels of each categorical attribute are encoded with a numeric value, allowing the categorical variable to be treated as continuous. Compared to an arbitrarily chosen pooling strategy, this approach is more data driven.

How numeric values are assigned in CatBoost encoding also factored in our decision. Encoding using target statistics such as the conditional mean (or even leave-one-out approaches) suffer from the problem of *target leakage* – that is, our encoding is based on the conditional distribution of the response in the training data. This may lead to overfitting on the training set, since we are using information from the response to predict the response.

To overcome the issue, a CatBoost encoder was fit on the training set (using the response labels). The encoder was then used to obtain encodings for the validation and training set (response labels from these sets were not used). This practice is similar to the common data standardisation. These CatBoost encoded datasets were then used to train the Decision Tree and LightGBM.

Feature Engineering for the Logistic Regression

The transformed data above has been enough for running all tree-based models below. However, for logistic regression, further feature engineering is performed as suggested by James et al. (2021)

To begin with, one assumption for the logistic regression is that the log-odds of subscription are linearly related to the predictors. To investigate this, we fit a logistic regression using the CatBoost encoded dataset above. We then obtained the predicted log-odds and plotted them against each continuous feature (Figure B.7, appendix). Variables such as **emp.var.rate** and **nr.employed** exhibit a linear association with the log-odds. Unfortunately, neither **campaign** nor **pdays** seemed linear with respect to the log-odds, prompting further processing.

pdays measures the number of days that passed since the last contact. Around 30,000 clients record a value of 999 (never been contacted), while those that were contacted, occupy values near 0. To process **pdays**, we took its inverse. Roughly speaking, values of 999 were converted to 0. Next, all nonzero values had their inverse taken. This resulted in a much better linear relationship between pdays and the log-odds of subscription.

For the remaining categorical variables, we log-transformed them (after incrementing zero values and changing sign of negative values in cons.conf.idx). As a result, campaign and pdays now fulfil the assumption of linearity much better, evident in figure B.8(see appendix).

Loss Matrix

With the same motivation as the store dataset, the ratio of clients not subscribing to the new term to those participating is 7.982. Hence, the assumed loss matrix is as followed:

ACTUAL\PREDICTED CLASS	POSITIVE	NEGATIVE
POSITIVE	**0**	**7.982**
NEGATIVE	**1**	**0**

Table B.15: loss matrix

Modelling

Decision tree

In the bank dataset, the base model chosen is decision tree. This is because there are many categorical predictors, which can be quickly encoded and fitted with the model. Also, the prediction rule is straightforward and easily interpretable. In short, it is simple enough to be the base knowledge for comparison with more sophisticated approaches to identify the key relationships between independent variables and dependent variable.

Therefore, the remaining categorical variables which have not been CatBoost encoded in feature engineering are simply dummy encoded. Next, all predictors in the training data are used to fit decision tree with a generous max_leaf_nodes=20. This is, however, susceptible to overfitting. Hence, cost complexity pruning technique is applied with the alpha hyperparameter obtained from GridSearch cross validation. The final obtained decision tree has less nodes than the initial fitting and is shown in figure B.9.

For the remaining model, all predictors from the training set are the inputs. This is because some of them involve slow learning and automatically select the best predictors for partition (e.g., tree-based boosting methods), whereas the others have been added regularisation term, so overfit with too many variables have been reasonably mitigated.

CatBoost

Given the presence of many categorical variables in the bank dataset, with some having high cardinality (as noted), we chose CatBoost as our first boosted tree method. CatBoost has a modified boosting

algorithm that mitigates data leakage. In its seminal paper (Prokhorenkova et al, 2018), CatBoost was also shown to outperform LightGBM and XGBoost on several datasets (using log-loss and zero-one loss). These reasons justify our choice for using CatBoost.

Table B.10 (see appendix) summarises the hyperparameters tuned for the CatBoost. Optuna was used for hyperparameter tuning. On each trial, a combination of these parameters was suggested, and a corresponding CatBoost model was trained. To evaluate the model, we used the cross-validation F1 score, and this was used as the objective for Optuna to maximise.

For more efficient computation, we utilised Pool datasets (these are datasets optimised for CatBoost). On each trial, the suggested weight for positive class was used to weigh observations in the Pool dataset; a model with the remaining suggested parameters was then trained on the Pool dataset.

Another trick we used to explore more combinations of parameters, was to set a learning rate of 0.05 when tuning the other hyperparameters. Once they were tuned, we lowered the learning rate to 0.01, and used early stopping (see next paragraph) to find the corresponding optimal number of boosting iterations. This trick is useful because a higher learning rate requires a lower number of boosting iterations, therefore reducing computation.

As mentioned, we utilised early stopping to find the optimal number of boosting iterations – this is different from the Bayesian optimisation approach utilised in the store dataset. Briefly, early stopping monitors the cross-validation score and stops when the metric improvement is trivial. 162 boosting rounds was suggested for the CatBoost.

LightGBM

The report follows the same procedure conducted previously. Specifically, Bayesian optimisation is used to tune hyperparameters and then the model is fitted on the training set. Hyperparemeter table can be viewed in table B.11 (see appendix)

L_2-penalised Logistic Regression

Similiar to the bank dataset, we tune the complexity parameter λ as well as the decision threshold τ. To tune λ, we again used grid search over 40 values in the range $[0.1, 30]$. The optimal λ is 2.14, which maximizes cross validation F1. We chose the L2 penalty since this consistently achieved higher cross-validation F1 than the L1 penalty.

We then fit an L2-regularised logistic model with λ = 2.14. Evaluating the cross-validation F1 for different values of the threshold τ (see Store dataset), the optimal outcome is $\tau = 0.199$.

Model stacking

Measuring the strengths of all four models considered with F_1 score, Lightgbm and CatBoost show dominant performances comparing to two simpler models logistic and decision tree. Hence, it is decided that a stack model is utilised on the two best performing approaches to improve the prediction power. For simplicity, the stacked meta-model chosen is L_2 penalised logistic regression. All hyperparameters tuned on lightgbm and CatBoost are maintained.

Variable importance consideration

Observing the variable importance figures B.12, B13, and B.14 obtained from decision tree, CatBoost and Lightgbm models, there are some notable points. First, some variables having good predictive power for

the output are repeatedly used by different models. For instance, continuous variables such as nr.employed and euribor3m often rank top in the figures. However, regarding categorical independent variables, different models utilise and identify information distinctively. For example, while decision tree evaluates month as the most crucial categorical predictors, CatBoost and Lightgbm study age and campaign more. This highlights the need to examine different sets of predictions rather than relying entirely on one model. Second, contrary to mutual information where continuous variables are at the top list for high association, some categorical predictors play more important roles in explaining the output. This may be partially explained by repeated information as observed in the correlation table and reinforce the power of machine learning where relationships cannot be analysed easily.

Finally, the logistic regression and stacking model do not have the function of variable important, but the coefficient plot does somewhat convey the same idea. Particularly, regarding the direction of relationships, month have a strong positive effect on the output, while euribor3m are among the predictors with highest negative associations.

Overall, notable important predictors are euribor3m, cons.price.idx, nr.employed, month, age, and campaign. Intuitively, it is reasonable as the current interest rate euribor3m would affect the investors' willingness to lend, whereas the remaining predictors determine whether clients have disposable income to subscribe a term deposit.

Model selection

Performance on validation set

To select the best model(s) for recommendation, the approach here is identical to the store dataset. Table B.15 summarises the performance of our models on the validation set. Similar to the store dataset, we consider metrics that take into account the class imbalance. We consider the F1 score, AUC and Average Loss (as calculated based on the Loss matrix).

Model	Error rate	Sensitivity	Specificity	Precision	Recall	F1 score	AUC	Average Loss
Decision tree (baseline)	0.101	0.271	0.978	0.607	0.271	0.374	0.777	5361.264
L_2--Penalized Logistic Regression	0.216	0.596	0.807	0.279	0.596	0.380	0.760	4256.502
LightGBM	0.126	0.564	0.913	0.447	0.564	0.499	0.791	3735.980
CatBoost	0.125	0.523	0.919	0.448	0.523	0.483	0.784	3976.332
Model Stacking	0.125	0.557	0.915	0.452	0.557	0.499	0.790	3764.872

Table B.15: model validation

The LightGBM and Model stack generally achieve the best validation F1, AUC and Average Loss. Specifically, they achieve the (joint) highest F1 of 0.499, the highest AUC values of 0.791 and 0.790 (respectively), as well as the lowest Average Loss of 3735.98 and 3764.872 (respectively).

While decision tree and logistic regression do outperform in some criteria, they do not strike a good balance in all metrics. For example, the decision tree has the lowest error rate and highest specificity and precision;

this is at the expense of having the poorest sensitivity and recall. On the other hand, the logistic regression has the highest sensitivity and recall, but it performs poorly on the F1, AUC and Average Loss.

Deployment considerations

The following discussion is based off Table S.14, which summarises aspects such as interpretability and maintenance cost of the models. As showed, the LightGBM and Model stack performed ideally in terms of the F1, AUC and Average Loss. Of the two however, the LightGBM is easier to interpret, and also requires less effort for maintenance, as it is simpler than the model stack. Hence, we select the LightGBM as our final model.

LighGBM may be less interpretable and harder to maintain than models such as the decision tree and the $\boldsymbol{L_2}$-penalised logistic regression. However, we believe that these challenges are outweighed by its superior predictive ability (Table S.14).

Model evaluation

Following previous practice in the store dataset, we now fit each model with the combined training and validation set. We then obtain performance metrics on the test set, to justify our choici fo final model based on generalisability to unseen data. Table B.16 summarizes these results.

Model	Error rate	Sensitivity	Specificity	Precision	Recall	F1 score	AUC	Average Loss
Decision tree	0.099	0.185	0.990	0.607	0.703	0.293	0.758	2520.384
L_2-Penalized Logistic Regression	0.144	0.514	0.899	0.279	0.389	0.443	0.761	1793.652
Light Gradient Boosting Machine	0.130	0.530	0.912	0.447	0.430	0.475	0.790	1705.760
CatBoost	0.129	0.475	0.920	0.448	0.427	0.450	0.773	1848.382
Model Stacking	0.130	0.514	0.915	0.452	0.430	0.468	0.788	1745.652

Table B.16: model evaluation

On the test set, Lightgbm again maintains decent performance compared to other models. For example, it still has the highest F1 of 0.475, the best AUC of 0.79, and the lowest average loss of 1705.760. These metrics on the test set justify our choice of LightGBM as final model.

Data Mining

Demographics such as marital status and job impact receptiveness

Further investigation into marital status and job is warranted, as these variables were found to be important in several models. Secondly, these data in practice are easy to obtain from clients, so the bank can utilise the following findings at ease.

To begin, we condense both variables to two categories only. **Job** is recategorized to indicate the unemployed (coded 1) and employed (coded 0). Similarly, for **marital**, 1 is coded for marriage (or divorce), and 0 for single. Next, the Fisher test is conducted with the following hypothesis: H_0: two variables are independent; H_1: two variables are related. With the p-value obtained of 0.0002, at 5% level of significance,

we reject the null hypothesis and conclude that these new formed variables are related. It is therefore now of interest to measure the interaction effect of these features on the output.

To evaluate the relationship between the output and these two variables, it is decided that the dependent variable should not be simply binary. Because this is too certain, there is not much information for learning. Instead, the odd of the output, a continuous variable, is used as it reveals more about the likelihood of the outcome, thus more rigorous information. Therefore, the two-way ANOVA for the odd and two formed variables are conducted with results:

	F-statistic	p-value	Coefficients
Intercept	2097.799	0.000	0.893
Condensed Job variable	486.696	0.000	1.352
Condensed Marital variable	141.205	0.000	-0.274
Interaction Job-Marital	28.674	0.000	-0.381
Mean odds	Employed	Unemployed	
Married (Divorced)	0.6188	1.5897	
Single	0.8925	2.2448	

Table B.17: Two-way ANOVA test

The p-value obtained suggest all coefficients are significant. There are some notable findings in the study. Particularly, people without a job are more likely to subscribe a term deposit, and the similar finding can be said for the single. Regarding the interaction, people without employment and have a married or divorced status are less likely to open the deposit as well.

From business perspective, the outcomes suggest some reasonable situations. Specifically, unemployed clients are more prone to financial insecurity and willing to have their money save up for contingency. Lower likelihood of married or divorced people to subscribe may be explained by the fact that their family finance is strong, or the other partner has already opened a saving account elsewhere.

With the findings, it is suggested that the bank focus their marketing concentration on the single or the unemployed as the target customers. They should receive more advertisement and attractive term package tailored for their needs (e.g., long-term mortgage, emergency fund) to increase the likelihood of subscribing, thus increasing the bank overall business value.

Market conditions are associated with customer receptiveness

From our modelling results, the economic variables are suggested to be the most important for modelling responsiveness to marketing campaigns. That is, they consistently have the highest variable importance (despite different variables prioritised in different models). In this section, we delve deeper into some chosen economic variables.

Employment

The (quarterly) average number of employed citizens is measured by **nr.employed**. This was found to be the most important variable for the CatBoost for predicting responsiveness to marketing campaigns (Figure B.12).

Intuitively, the more people employed, the higher the average wage, and therefore the more savings generated. This would lead to a larger propensity to subscribe to a term deposit, to accumulate the savings. Therefore, we hypothesize:

H1: The higher the total employment, the more willing customers will be to subscribe to a deposit

This hypothesis is supported by coefficients from the logistic regression (Figure B.14, see appendix). **Nr.employed** was found to have an estimated coefficient of close to 1. This means that holding all other factors constant, an increase in the **nr.employed** by 1 will lead to an estimated increase of 1 in the log-odds of subscription (or equivalently, an increase in the subscription odds by a factor of 0.36).

Therefore, there is suggestion that customers who are contacted during a period of larger national employment, will be more willing to subscribe to a term deposit. The bank could therefore aim to increase the number of marketing campaigns during such periods, to attract a greater number of subscriptions.

Consumer Price index

Cons.price.idx measures the monthly average Consumer Price Index (CPI). This is the weighted average price from a basket of goods (the goods chosen to represent typical spending in households) (Australian Bureau of Statistics, 2010). **Cons.price.idx** was found to be the variable most strongly associated with subscription in the logistic regression; it is also moderately important for the LightGBM and CatBoost, based on feature importance.

The CPI is commonly used as a measure of inflation; but for our purposes, it may also be viewed as a proxy for how much households are spending on goods (since, the more households spend, the more prices will be inflated – i.e., the higher the CPI). Hence, we hypothesise that:

H2: the higher the CPI, the more willing customers are to subscribe to a deposit.

In some sense, this is reasonable: larger (household) expenditure on goods is indicative of larger wages. And with larger wages comes greater savings, and therefore an increased desire to subscribe to long-term deposits to store those savings.

This hypothesis is supported by results from the logistic regression (Figure B.14, see appendix). The estimated coefficient for **cons.price.index** is around 3. This means that holding all other factors constant, an increase **in cons.price.index** by 1 will lead to an estimated increase of 3 in the log-odds of subscription (or equivalently, an increase in the subscription odds by a factor of 20).

Market conditions such as employment and the CPI are suggested to be influential for customer responsiveness to marketing campaigns. Yet, these factors are outside direct control of the bank and the client. However, there are some steps the bank may take considering the insights.

For example, since higher employment and CPI are associated with greater customer receptiveness, the bank may leverage this, by performing more marketing campaigns during these periods. This would result in more term deposits subscribed, and larger profit generated for the bank.

Conversely, during periods of low employment or low CPI, the bank may cut back on expenditure for marketing campaigns, to adjust to the decreased demand for term deposits.

REFERENCES

James, G. et al., 2021. In An introduction to statistical learning: With applications in R. New York, NY: Springer, pp. 22, 107, 198–199.

The Australian Bureau of Statistics, 2010. “Consumer Price Index”. https://www.abs.gov.au/AUSSTATS/abs@.nsf/DSSbyCollectionid/1E564CACF4CBEC32CA256ED8007EF06E?opendocument#:~:text=The%20price%20of%20the%20CPI,the%20index%20would%20read%20135.0., accessed 20th May 2022.

Prokhorenkova1 et al., 2018. *CatBoost: unbiased boosting with categorical features*. From the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

APPENDIX

Store dataset

CONTINUOUS		**DISCRETE**	**BINARY**
MON	PCOLLSPND	FRE	CC_CARD
AVRG	AMSPEND	PC_CALC20	VALPHON
PSWEATERS	PSSPEND	PROMOS	WEB
PKNIT_TOPS	CCSPEND	DAYS	RESP
PKNIT_DRES	AXSPEND	CLASSES	
PBLOUSES	TMONSPEND	COUPONS	
PJACKETS	OMONSPEND	STYLES	
PCAR_PNTS	SMONSPEND	STORES	
PCAS_PNTS	PREVPD	MAILED	
PSHIRTS	GMP	RESPONDED	
PDRESSES	FREDAYS	CLUSTYPE	
PSUITS	MARKDOWN		
POUTERWEAR	RESPONSERATE		
PJEWELRY	HI		
PFASHION	LTFREDAY		
PLEGWEAR	PERCRET		

Table S.1: Catalogue of variables by datatype

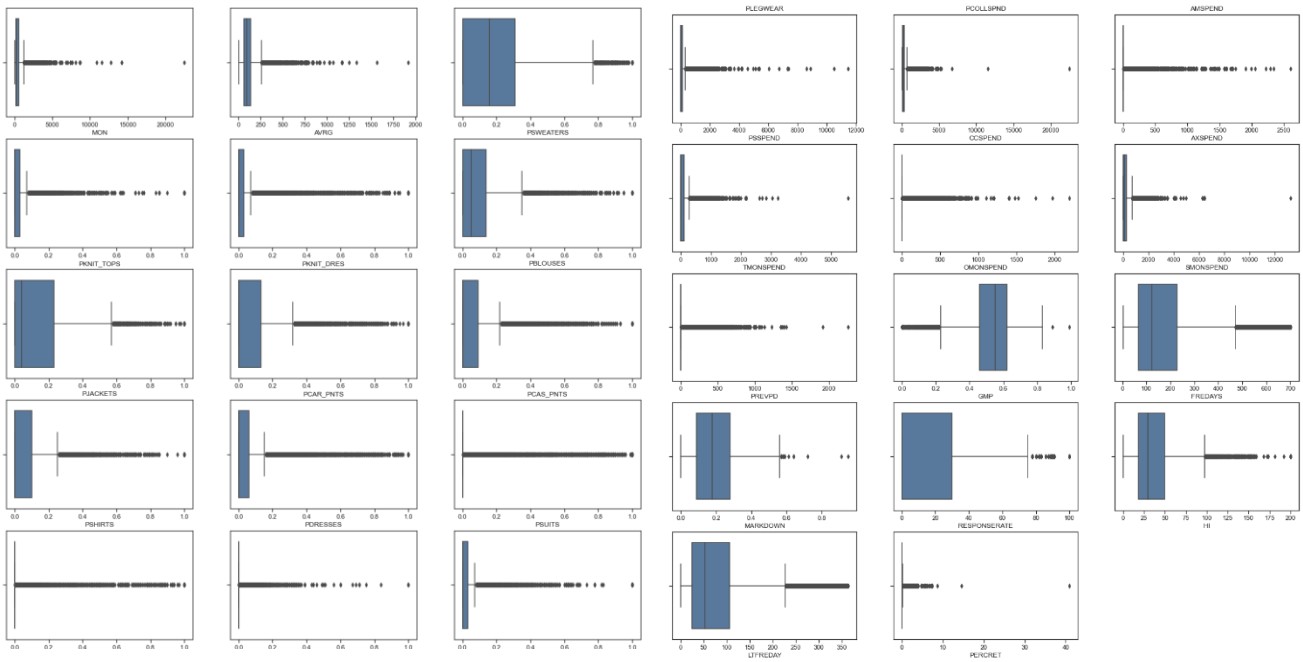


Figure S.1: Box plot for continuous variables

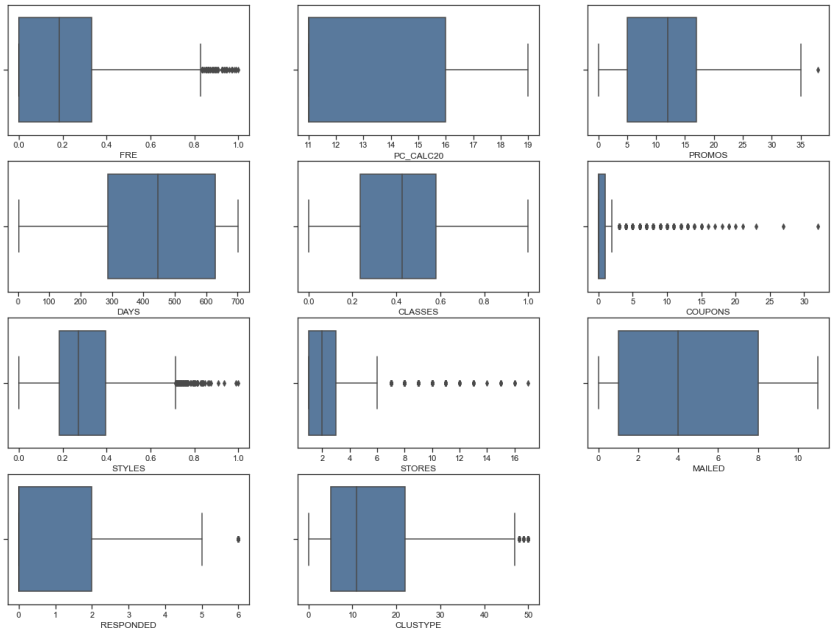


Figure S.2: box plot for discrete variables

Variable	Skewness	Kurtosis	Variable	Skewness	Kurtosis
MON	0.281285225	-0.049844752	PCOLLSPND	3.049536188	9.776953989
AVRG	3.739644381	32.08316291	AMSPEND	32.18505762	1756.475057

PSWEATERS	1.496871135	2.231835267	PSSPEND	8.327310789	139.2891499
PKNIT_TOPS	6.224721698	61.49457275	CCSPEND	11.05804686	388.5428133
PKNIT_DRES	4.493538199	25.74975748	AXSPEND	9.314068039	122.8414032
PBLOUSES	2.693374851	10.61111005	TMONSPEND	5.556261137	73.42402618
PJACKETS	1.563927373	2.561251948	OMONSPEND	5.839383098	57.70568001
PCAR_PNTS	2.420736588	7.737642179	SMONSPEND	-0.535525204	-1.310449772
PCAS_PNTS	3.044525805	12.00312993	PREVPD	4.686352493	37.95434344
PSHIRTS	2.887088494	12.0102508	GMP	-1.35420412	2.155960695
PDRESSES	3.545413637	14.60637925	FREDAYS	-0.199664837	-0.29071961
PSUITS	5.029741699	27.8824238	MARKDOWN	-0.303976858	-0.507188326
POUTERWEAR	7.219709584	57.34883593	RESPONSERATE	1.534864985	1.806483345
PJEWELRY	10.16551754	173.8376073	HI	1.737830579	3.984408719
PFASHION	5.997587506	53.42092406	LTFREDAY	-0.162536133	-0.417745878
PLEGWEAR	10.7974036	167.8853108	PERCRET	38.1860835	2862.675859

Table S.3: Skewness and kurtosis for continuous variables

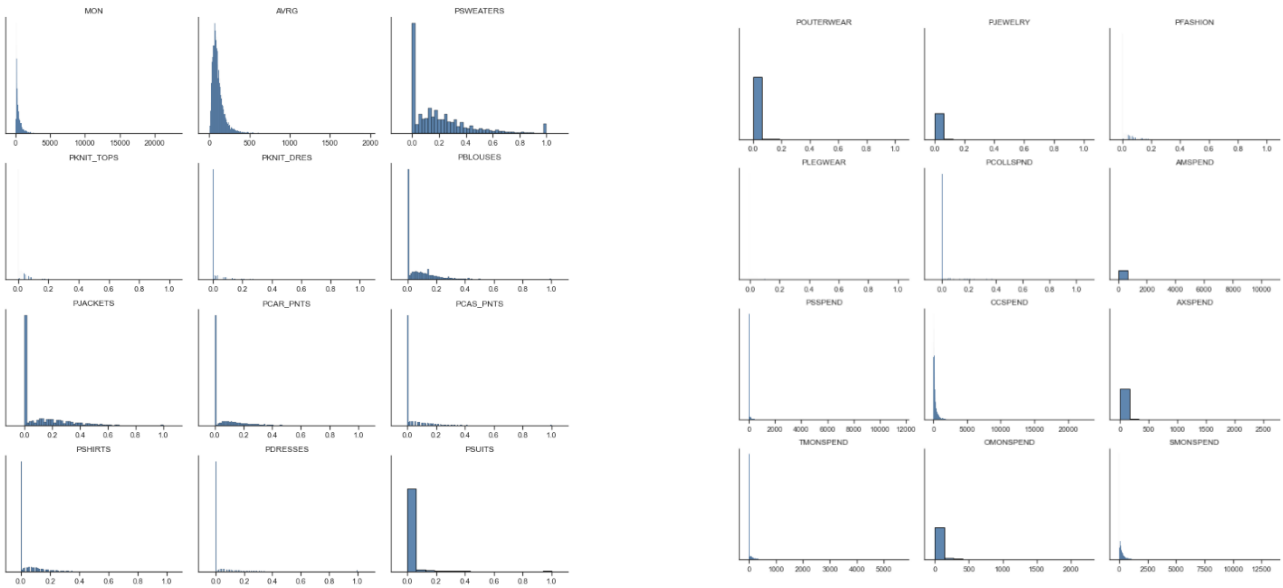


Figure S.3: distribution plot for continuous variable

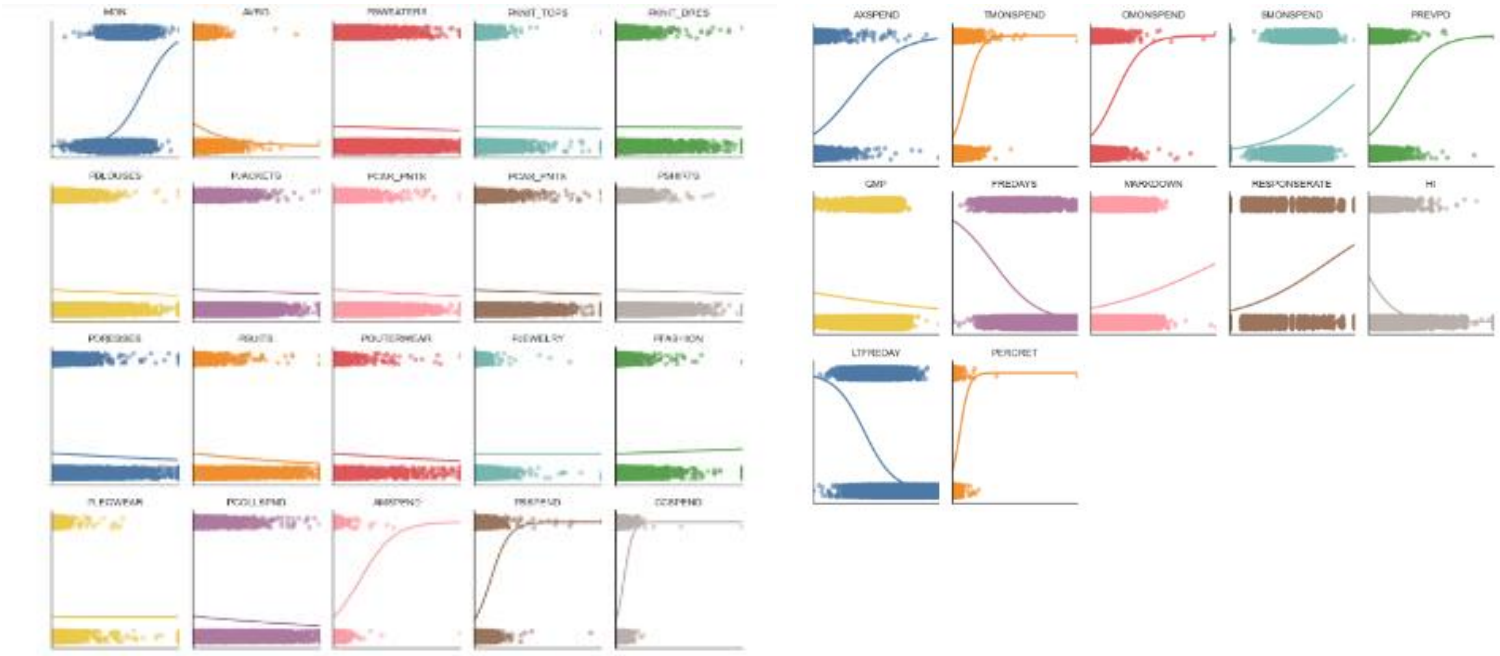


Figure S.5: Sigmoid plots for continuous variables

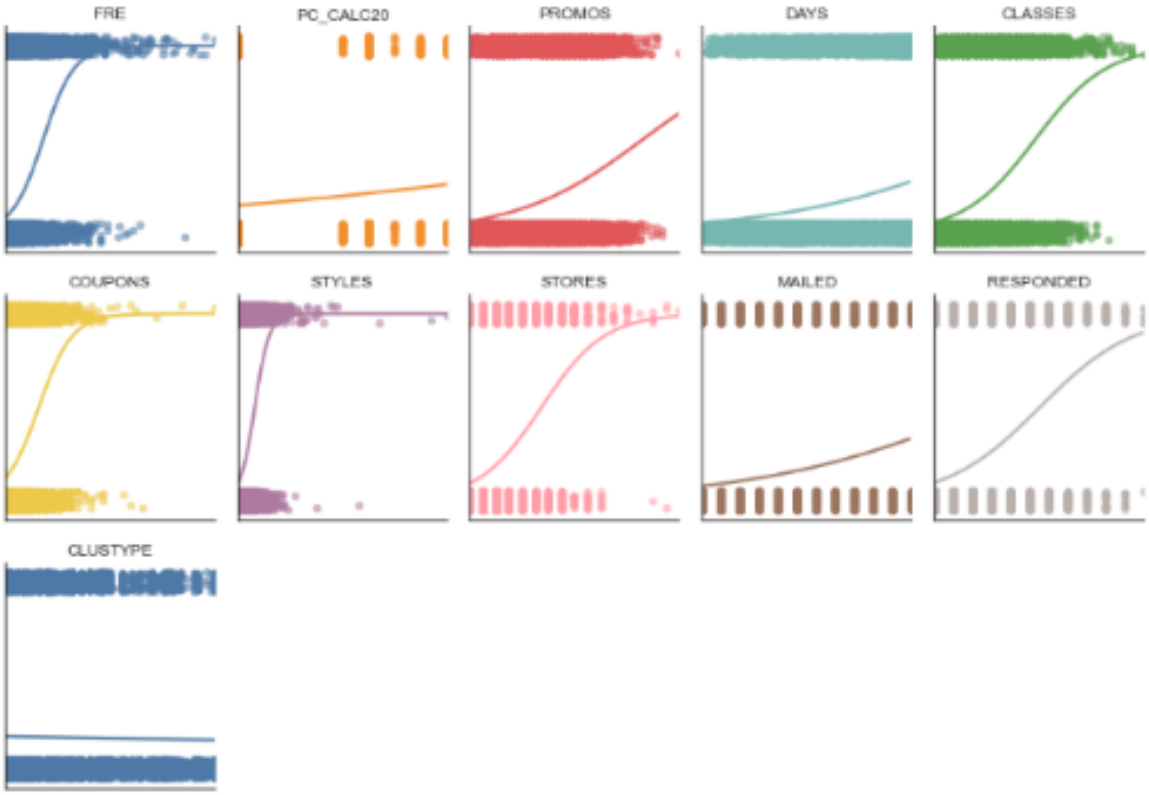


Figure S.6: Sigmoid plots for discrete variables

	Mutual Information
FRE	0.081
STYLES	0.076
CLASSES	0.062
RESPONDED	0.052
COUPONS	0.044
DAYS	0.043
STORES	0.04
PROMOS	0.027
MAILED	0.023
PC_CALC20	0.004
CLUSTYPE	0.002

	Mutual Information		Mutual Information
LTFREDAY	0.109	**PJACKETS**	0.027
FREDAYS	0.064	**PSSPEND**	0.024
MON	0.053	**AVRG**	0.024
SMONSPEND	0.05	**OMONSPEND**	0.023
RESPONSERATE	0.049	**PKNIT_DRES**	0.022
HI	0.042	**PSWEATERS**	0.021
TMONSPEND	0.035	**PREVPD**	0.02
CCSPEND	0.034	**PCOLLSPND**	0.018
PBLOUSES	0.031	**PJEWELRY**	0.016
PDRESSES	0.031	**PLEGWEAR**	0.015
PERCRET	0.031	**MARKDOWN**	0.014
PCAR_PNTS	0.029	**POUTERWEAR**	0.012
PCAS_PNTS	0.029	**PSUITS**	0.009
PSHIRTS	0.028	**GMP**	0.009
PFASHION	0.028	**AXSPEND**	0.007
PKNIT_TOPS	0.027	**AMSPEND**	0.001

Table S.5: mutual information for numerical variable

VARIABLE	MUTUAL INFORMATION
RESPONDED_1	0.000
RESPONDED_2	0.002
RESPONDED_3	0.005
RESPONDED_4	0.006
RESPONDED_5	0.005
RESPONDED_6	0.017

Table S.7: mutual information for one hot encoded RESPONDED

VARIABLE	MUTUAL INFORMATION
SPENDING_PLEGWEAR	0.061
SPENDING_PSWEATERS	0.06
SPENDING_PJEWELRY	0.058
SPENDING_PKNIT_TOPS	0.058
SPENDING_PBLOUSES	0.058
SPENDING_PKNIT_DRES	0.058
SPENDING_PCAS_PNTS	0.057
SPENDING_PSUITS	0.057
SPENDING_PSHIRTS	0.057
SPENDING_POUTERWEAR	0.057
SPENDING_PDRESSES	0.056
SPENDING_PCAR_PNTS	0.055
SPENDING_PFASHION	0.055
SPENDING_PCOLLSPND	0.055
SPENDING_PJACKETS	0.051

Table S.9: mutual information for spending relevant variables

	param_penalty	param_C	param_solver	mean_test_score	std_test_score
0	l2	0.033333333	saga	0.489203177	0.013104167
22	l2	0.076142132	saga	0.502467316	0.011931185
23	l2	0.080862534	saga	0.502813246	0.012211641
24	l2	0.086206897	saga	0.502887485	0.012096641
25	l2	0.092307692	saga	0.503963845	0.012813069
26	l2	0.099337748	saga	0.505484015	0.012522033
27	l2	0.107526882	saga	0.507307241	0.012496791
28	l2	0.1171875	saga	0.508325216	0.012878053
21	l2	0.071942446	saga	0.501052858	0.01130457
29	l2	0.128755365	saga	0.508930836	0.011683299
31	l2	0.160427807	saga	0.5104351	0.011515552
32	l2	0.182926829	saga	0.511098457	0.010773454
33	l2	0.212765957	saga	0.510306699	0.010569923
34	l2	0.254237288	saga	0.510288514	0.011555194
35	l2	0.315789474	saga	0.511830062	0.012101044
36	l2	0.416666667	saga	0.512517785	0.012176749
37	l2	0.612244898	saga	0.511941747	0.011226187
30	l2	0.142857143	saga	0.510493531	0.011493304
20	l2	0.068181818	saga	0.50057839	0.010910045
19	l2	0.064794816	saga	0.500472641	0.011210602
18	l2	0.061728395	saga	0.500069421	0.010984491

1	l2	0.034207526	saga	0.489772619	0.012769962
2	l2	0.035128806	saga	0.491026825	0.012820121
3	l2	0.036101083	saga	0.491134638	0.012908765
4	l2	0.037128713	saga	0.491699424	0.012677147
5	l2	0.038216561	saga	0.491412825	0.012239765
6	l2	0.039370079	saga	0.491982895	0.012077043
7	l2	0.040595399	saga	0.492443042	0.011961247
8	l2	0.041899441	saga	0.493243288	0.011633362
9	l2	0.043290043	saga	0.495185348	0.011907567
10	l2	0.044776119	saga	0.496196569	0.011725777
11	l2	0.046367852	saga	0.497003936	0.01130696
12	l2	0.048076923	saga	0.497694972	0.012078031
13	l2	0.049916805	saga	0.497805234	0.011301113
14	l2	0.051903114	saga	0.49831374	0.010389398
15	l2	0.054054054	saga	0.498408443	0.010366754
16	l2	0.056390977	saga	0.498750578	0.009710444
17	l2	0.058939096	saga	0.499263687	0.011013531
38	l2	1.153846154	saga	0.512463646	0.009459626
39	l2	10	saga	0.514683179	0.009302428

Table S.16: Grid search for penalty hyperparameter

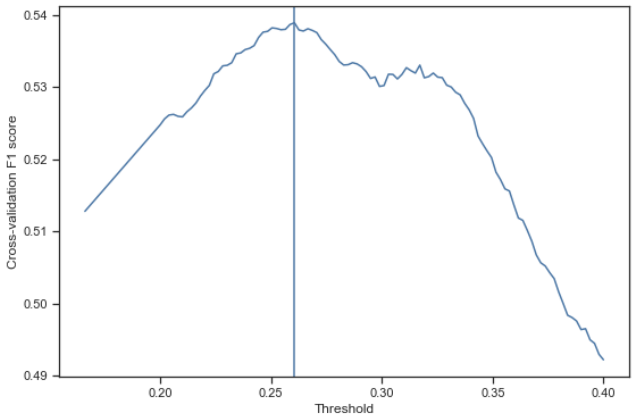


Figure S.7: Cross validation F1 score for different levels of the threshold parameter

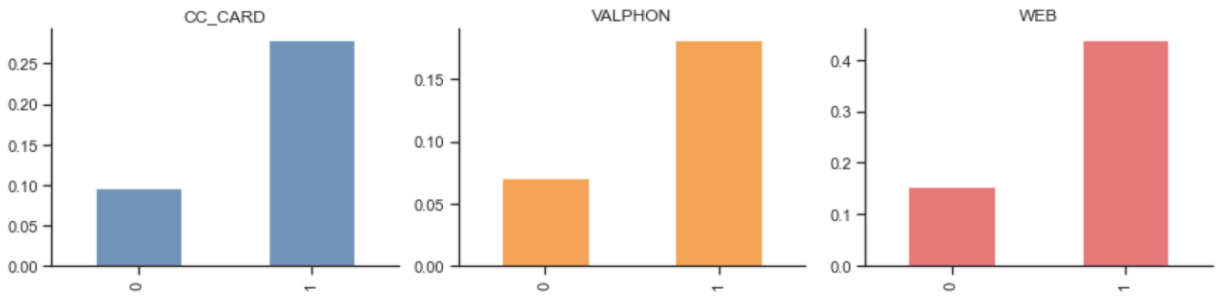


Figure S.17: cross tabular plot for binary variable

HYPERPARAMETER	TYPE	POSSIBLE VALUES	TUNED VALUE
Number of trees	Integer	200	200
Number of leaves	Integer	[10, 20]	20
Minimum number of examples	Integer	[30, 50]	33
Objective function	Category	[Gini-index,Entropy]	Gini-index
Positive weight	Float	[2,5]	2.829

Table S.12: Summary of Bayes optimization's input and output for random forest

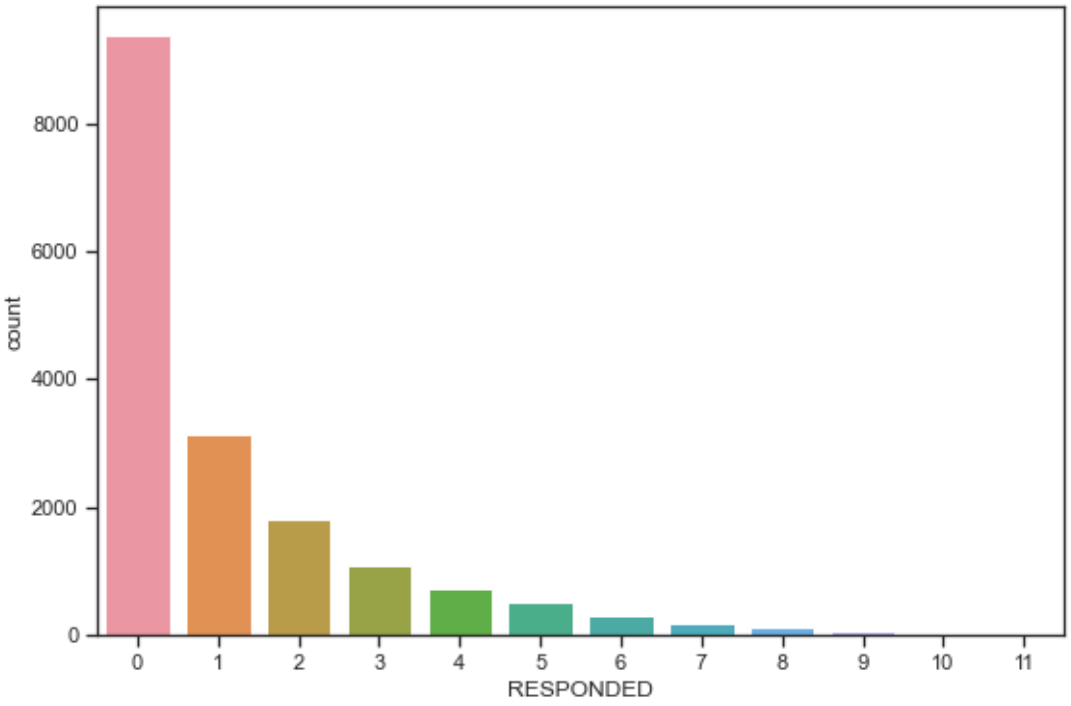


Figure S.18: count plot for **RESPONDED**

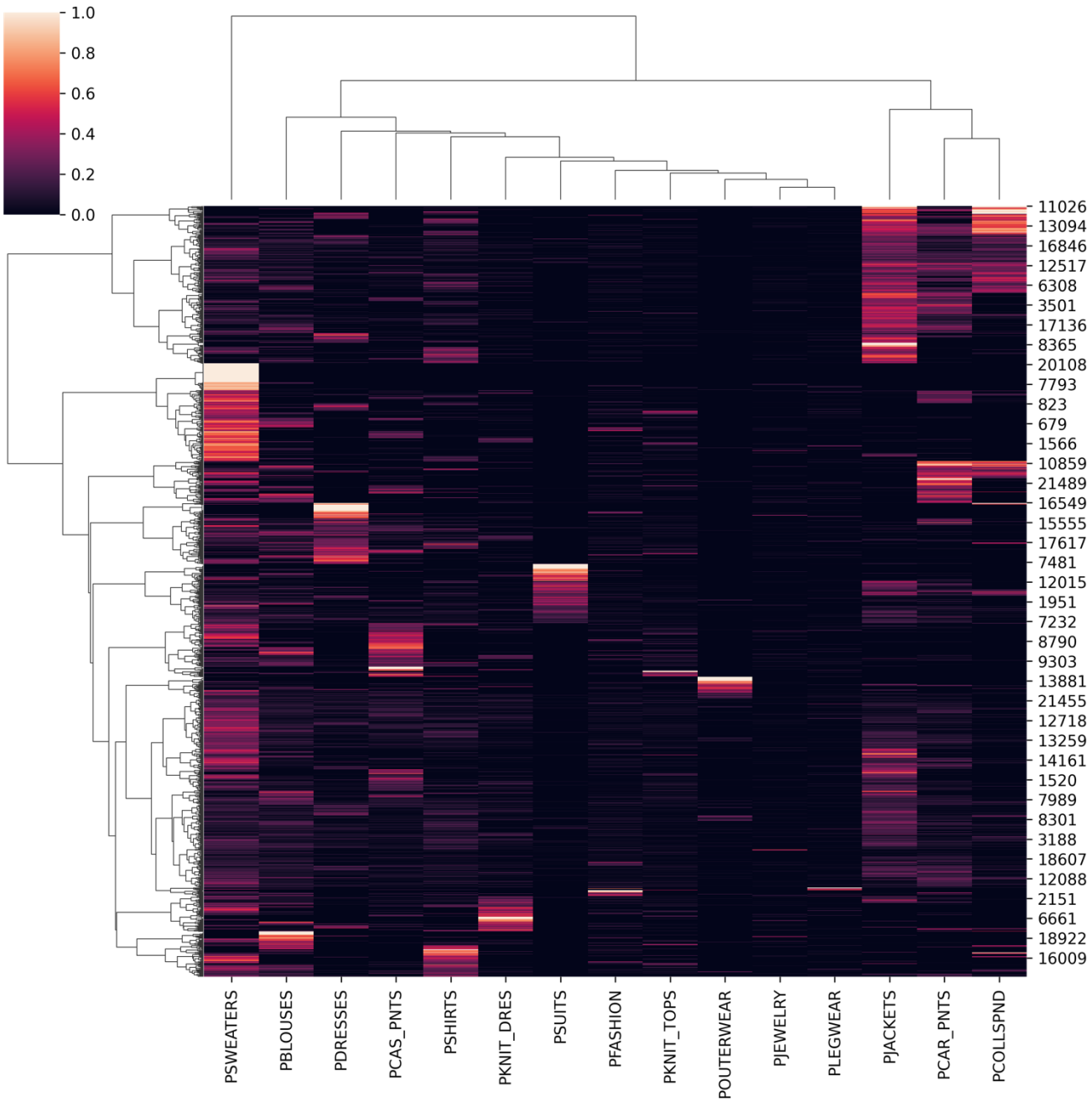


Figure S.16: Clustering of customers from the store dataset, using the proportions spent in different clothing categories (PSWEATERS, PBLOUSES etc.). Cutting the row dendrogram to give three clusters, we see that customers can be clustered into three groups: those which spend a large proportion on jackets, class career pants, and collectible line clothing; those which spend a lot on sweaters, and other customers.

Bank Appendix

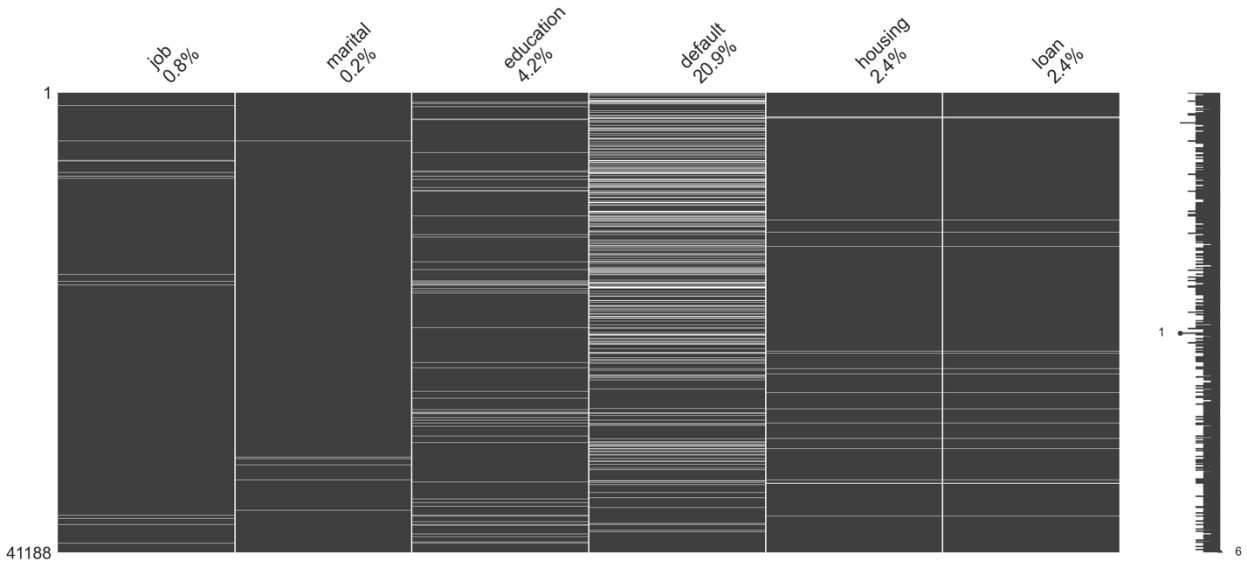


Figure B.1: Missingness plot for the variables **job, marital**, **education**, **housing** and **loan**.

Figure B.2: Boxplots for the continuous variables

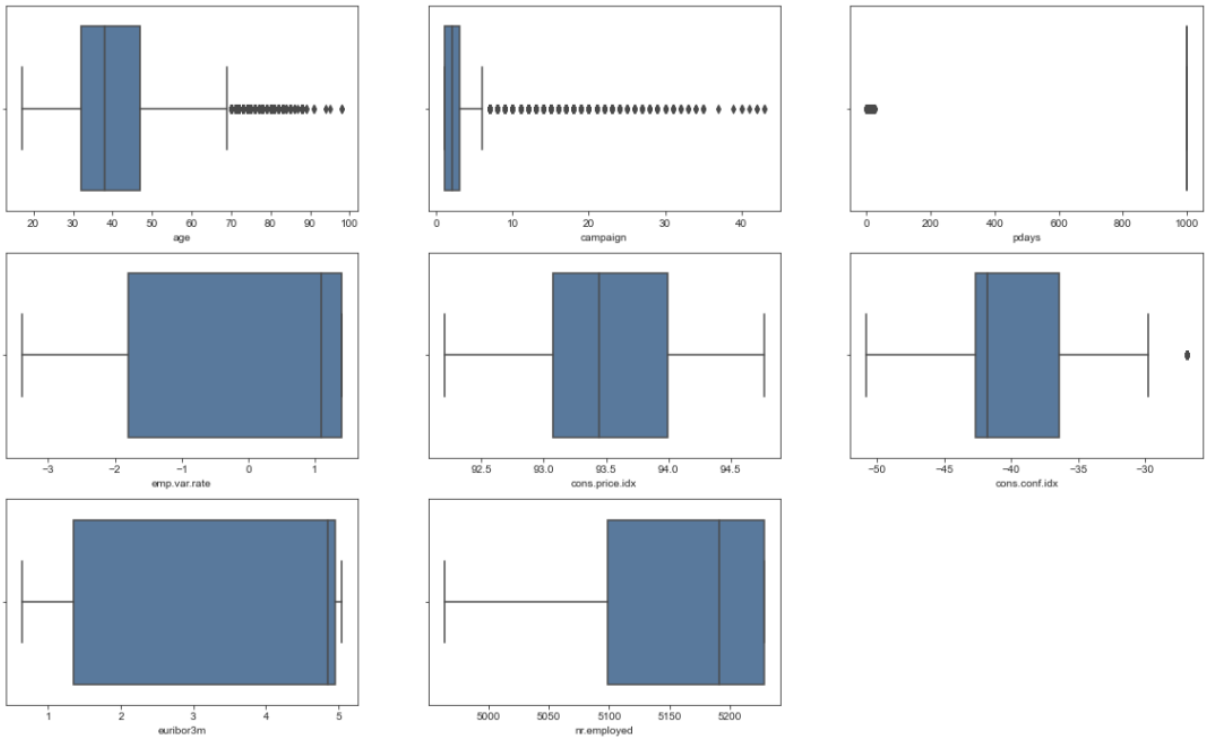


Figure B.3: Sigmoid functions relating each continuous attribute to the log odds of response, as obtained by fitting individual logistic regression models.

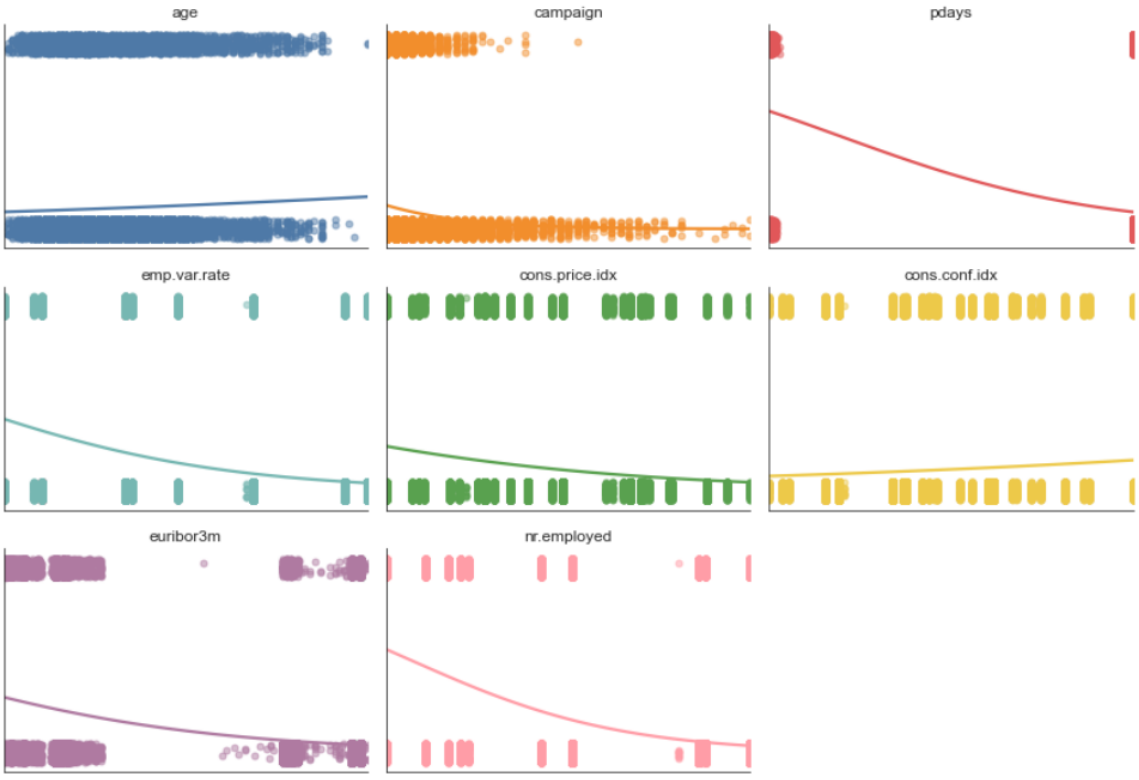


Figure B.4: correlation heatmap of all continuous variables

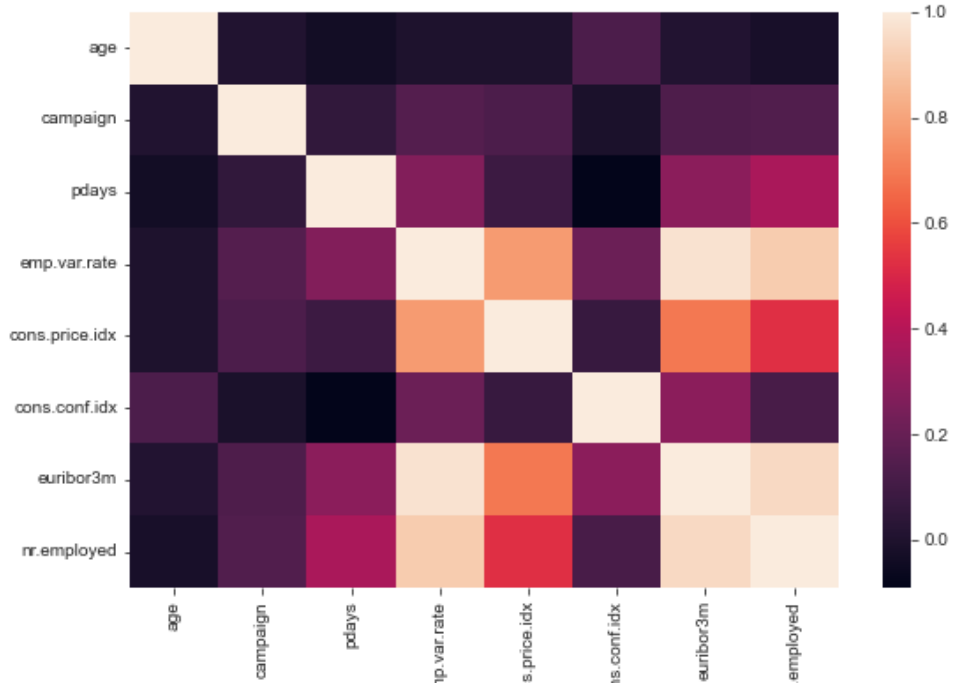


Figure B.5: crosstab plots for each categorical variable.

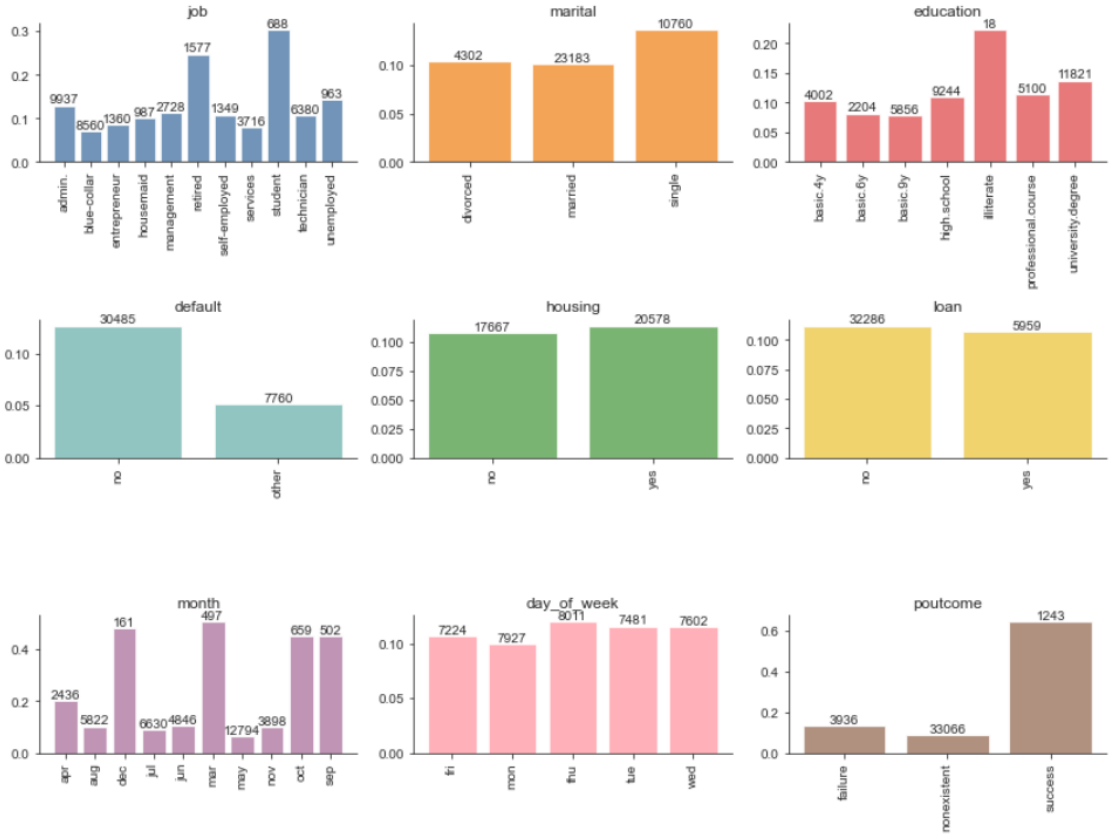


Table B.7 Mutual information for continuous and categorical variables

Continuous variables	**Mutual Information**
euribor3m	0.071
cons.price.idx	0.066
cons.conf.idx	0.066
nr.employed	0.063
emp.var.rate	0.055
pdays	0.037
age	0.01
campaign	0.006
Categorical variables	**Mutual Information**
poutcome	0.029
month	0.026
contact	0.011
job	0.009
default	0.005
education	0.002
marital	0.001
day_of_week	0
housing	0
loan	0

Figure B.7: log odds of response, against each continuous variable. An assumption in logistic regression is that the log-odds are relatively linear with the predictors. This assumption is not fulfilled well for **campaign** or **pdays**.

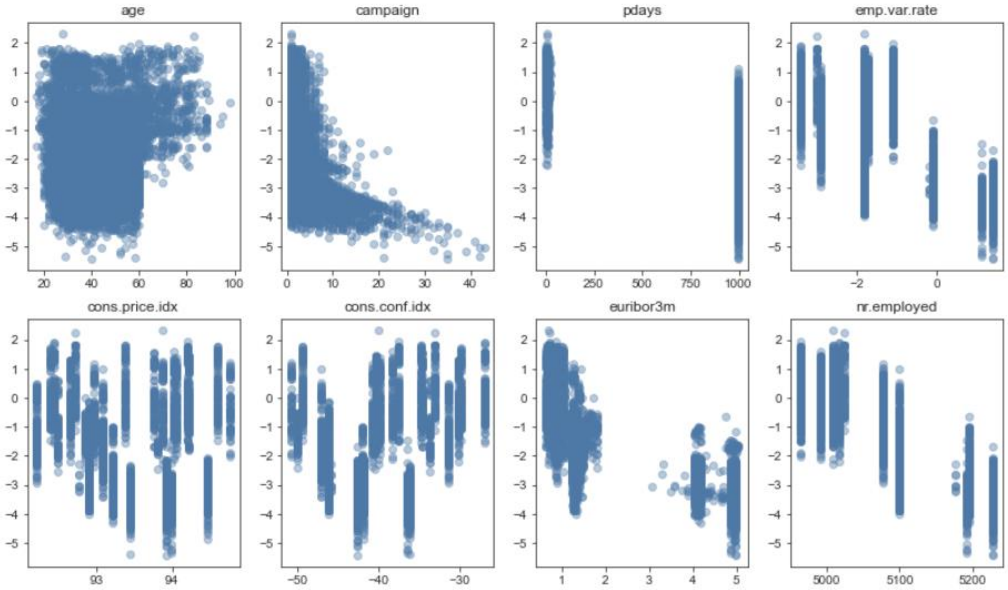


Figure B.8: log odds of response, against each continuous variable, after processing. Now, **campaign** and **pdays** seem to have a stronger linear association with the log odds of response.

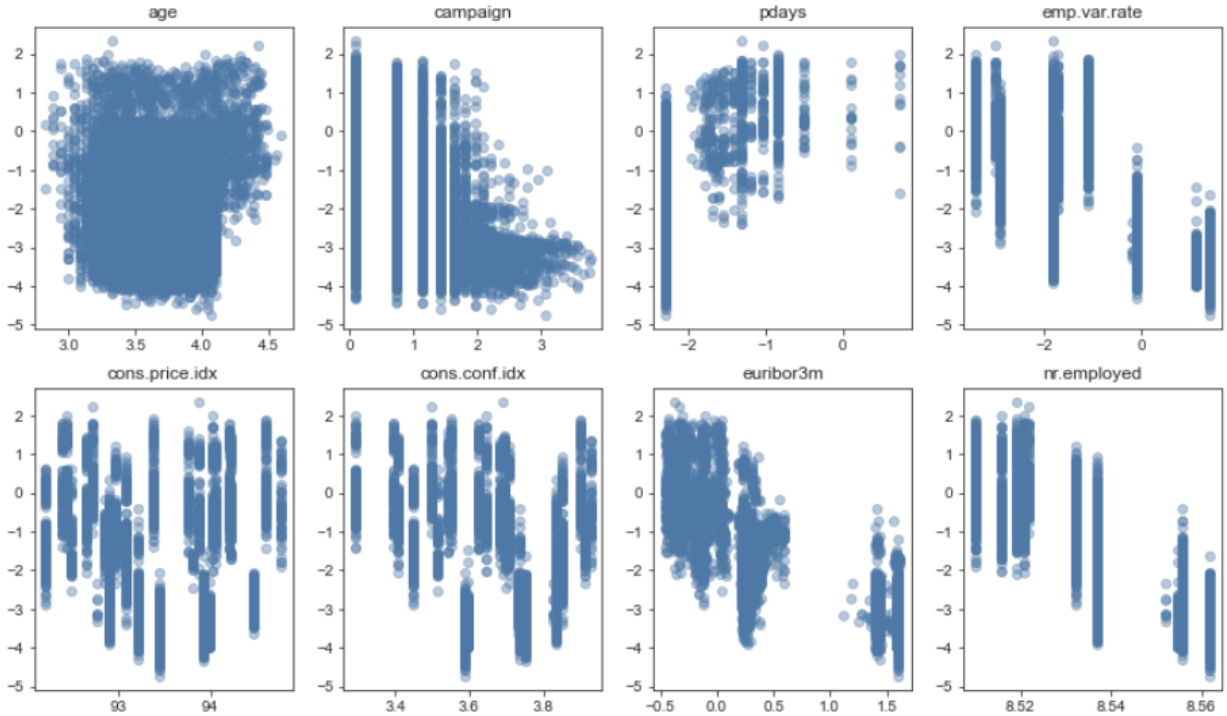
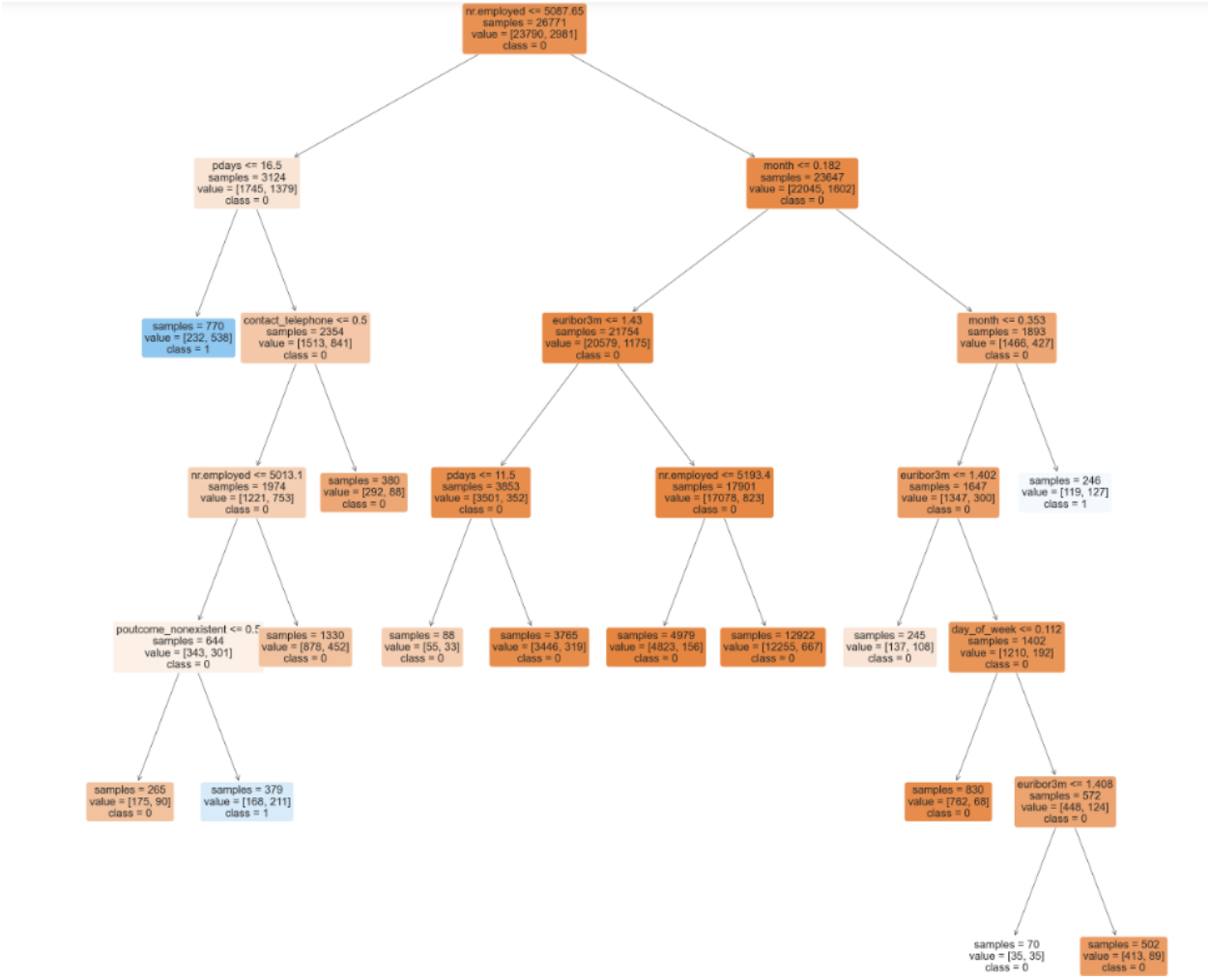


Figure B.9 the estimated Decision tree



Hyperparameter	Type	Possible values	Tuned value
Weight for positive class	continuous	[1,10]	3.256
Depth of each tree	integer	[4,10]	6
Proportion of data for bagging	continuous	[0.5, 1]	0.619
Proportion of features considered at each split	continuous	[0.5, 1]	0.763
L2 regularisation term for the cost function	continuous	[1e-8, 10]	0.916
Minimum number of samples in each leaf	integer	[1,128]	26

Table B.10: hyperparameter optimisation for the CatBoost

HYPERPARAMETER	TYPE	POSSIBLE VALUES	TUNED VALUE
Weight for positive class	Continuous	[1,10]	3.907
Proportion of data for subsample	Continuous	[0.4,1]	0.791
Maximum number of leaves	Integer	[5,40]	19
Minimum number of examples	Integer	[5,50]	25

Table B.11**:** hyperparameter optimisation for the LightGBM

Figure B.12 CatBoost Variable Importance

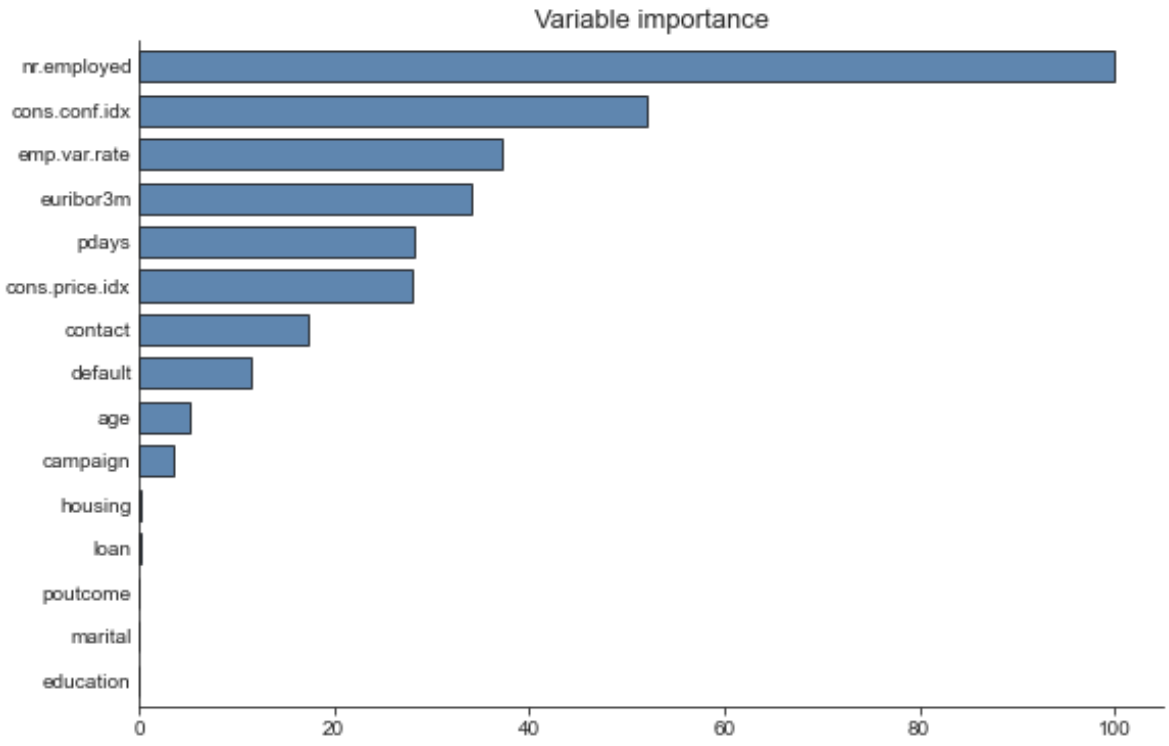


Figure B.13 LightGBM Variable Importance

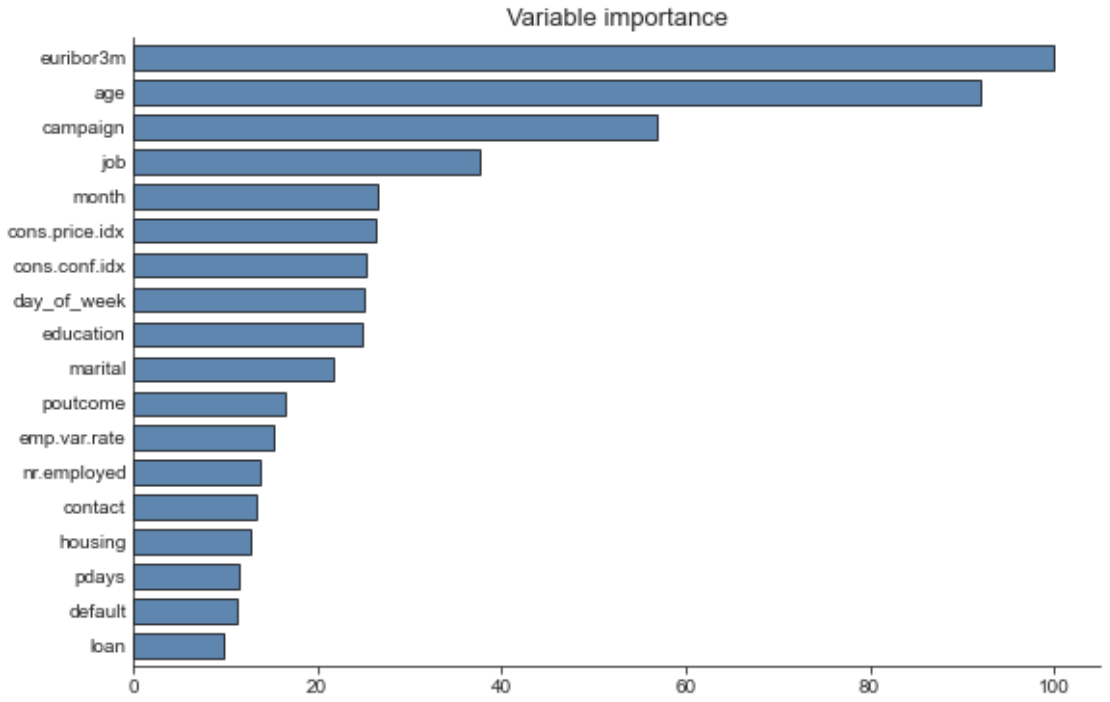


Figure B.14 logistic regression coefficients plot