QBUS6810 report

- Classification Project
- Group 107

460041274

460298498

510295538

500194164

510228891

Table of contents

1. Introduction

With the development of information technology, it has become an indispensable process for businesses operating at present to help enterprises make effective development plans through big data analysis. To help a bank and a fashion store improve the effectiveness of their marketing campaigns, this report aims to develop an effective statistical learning model to help customers explore and find the key factors for the success of their marketing activities, and use this to predict the success probability of customer marketing activities in the future.

Firstly, this report applied exploratory data analysis (EDA) to identify natures and insights about the data. Secondly, it adopted necessary feature engineering techniques to transform the two datasets of Bank and Store based on their natures. Thirdly, machine learning models were developed to predict the response variable of the two datasets. A series of hyperparameter tuning and feature importance analyses were carried out. Finally, based on the model and analysis results, this paper summarized three insights for each client to help their management make real-world and efficient decisions.

2. Problem formulation

In fact, although our bank and fashion store clients have been sending emails for marketing campaigns to many consumers. However, there is still room for improvement in the response rate of consumers to such marketing activities. This report will deeply analyze and explore the background and behavior information of consumers to help customers achieve a higher activity response rate. Based on these predictions, banks and fashion stores can allocate their marketing resources more effectively to maximize the company's sales.

For the report, it will use some machine learning models as Gradient Boosted decision trees (GBDT), Logistic, Lasso and Ridge models, etc. to help the bank predict whether the customer is more likely to subscribe to the bank's fixed deposit service and help the fashion store predict whether the customer is likely to respond to it Promotional emails.

3. EDA

3.1 Bank

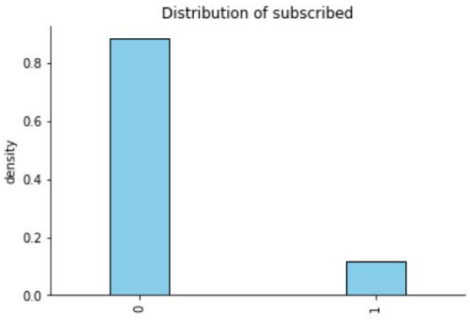
3.1.1 Data overview

The 'bank_train' and 'bank_test' contain 29,387 and 15,824 observations of customer respectively. Each dataset contains 15 predictors, and the train dataset contains a response variable, "subscribed". Of the 15 variables, we allocate them into datatype in accordance with the properties of the date themselves. Even though discrete variable belongs to quantitative data by its nature, it was treated was categorical data in machine learning. The result is presented as table on the right. The distributions of unprocessed data were presented below:

Variables	datatype_level 1	datatype_level 2
age	continuouos	quantitative data
balance		
pdays		
job	nominal	categorical data (qualitative data)
marital		
education		
month		
contact		
poutcome		
previous	discrete	
day		
campaign		
default	binary	
housing		
loan		
subscribed	response	

- **Response variables:**

The response variable is a binary variable, consisting of “yes” and “no”. It was transformed by dummy encoding, where 0 presents not been subscribed, which are negative cases, 1 presents positive cases. We can find that only 11.70% of all observations were positive, which means the success rate of the bank’s marketing campaign was low.



no	0.883009
yes	0.116991

- **Quantitative variables – density distribution:**

Appendix 1 presents the density distribution of quantitative variables. Of these variables, `pdays`, `previous`, and `campaign` has a much higher deviation coefficient than `age`. Moreover, `pdays` values from a wide range, but seems extremely concentrated on small values, which is considered abnormal. Thus, to find out the reason, we did more detailed work on the feature and find that there are 50 unique values ranging between [-1,854], and the head 10 frequency values in `pdays` is presented in the right place. Form the data frame, `-1` takes approx. 82% of the variable. By combining the metadata, we find that `-1` is actually a nominal predictor, which presented the clients was previously not contacted, but other values are numeric predictors, presenting a number of days. Therefore, variable `pdays` contains 2 types of data, which might negatively affect the performance of algorithm models.

	pdays	frequency	frequency_percent
0	-1	24036	0.8179
1	92	99	0.0034
2	182	95	0.0032
3	91	82	0.0028
4	183	77	0.0026
5	181	77	0.0026
6	370	60	0.0020
7	184	58	0.0020
8	94	55	0.0019
9	189	48	0.0016

Also, `balance` seems has an extremely high kurtosis of its density distribution. By observing the frequency box on right, we find the frequency of 0 is 21 times higher than the second-highest frequency term, which is also abnormal. Thus, we also will treat 0 in balance as a missing value.

	frequency	fre_percent
0	2348	0.084103
1	135	0.004836
2	110	0.003940
4	86	0.003080
3	86	0.003080
...	...	...
-694	1	0.000036
4641	1	0.000036
2715	1	0.000036
2594	1	0.000036
3947	1	0.000036

- **Continuous features – Measure of dependence:**

The dependence measure of continuous features with the response variable was done via the regression plot from **Appendix 2**. It can be observed that `balance`, `previous`, `pdays` are positively related to the response variable. Of these variables, `age` is a weak positive associated with the response variable. However, `day` and `campaign` seems to have little or no correlation with the response variable.

- **Qualitative features:**

Both discrete variables, `previous` and `campaign` were also explored like categorical variables in machine learning, each with over 25 unique values. The further transformation will be provided in order to prevent the curse of dimensionality. Please

refer to Appendix 3 for specific data distribution.

3.1.2 Missing Values problem

By applying method .info(), the dataset presents complete information for all variables. Thus, there are no empty (Nan or NaN) values in the dataset. The same situation also exists in the test dataset.

```
#   Column      Non-Null Count  Dtype
---  ------      --------------  -----
0   age         29387 non-null  int64
1   job         29387 non-null  object
2   marital     29387 non-null  object
3   education   29387 non-null  object
4   default     29387 non-null  object
5   balance     29387 non-null  int64
6   housing     29387 non-null  object
7   loan        29387 non-null  object
8   contact     29387 non-null  object
9   day         29387 non-null  int64
10  month       29387 non-null  object
11  campaign    29387 non-null  int64
12  pdays       29387 non-null  int64
13  previous    29387 non-null  int64
14  poutcome    29387 non-null  object
15  subscribed  29387 non-null  object
dtypes: int64(6), object(10)
```

Through the bar charts of all qualitative data (**Appendix 3**), we find there are unknown quantities from the following variables - `job`, `education`, `contact` and `poutcome`. We will provide a detailed summary of all missing variables in the training set in later paragraphs.

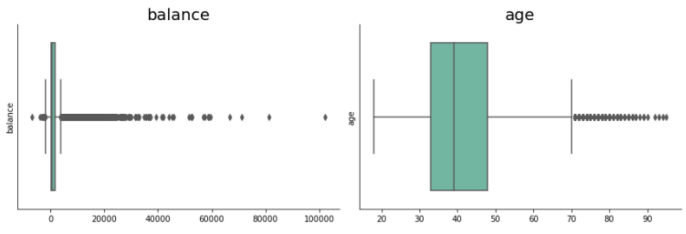
As shown in the right plot, `job` and `education` only contains less than 5% of `unknown` values. The situation in `contact` and `poutcome` is more complex. We will try different methods to settle the unknown quantities. These will be also discussed in a later part.

	var	missing_cnt	percentage
0	job	189	0.01
1	education	1189	0.04
2	contact	8364	0.28
3	poutcome	24040	0.82

3.1.3 Outliers problem

According to the distribution of continuous data, and frequency of qualitative data, we will mainly explore outliers in the following variables, which are `balance`, `age`.

Of these two variables, the boxplots are shown on right. Because both variables are not normally distributed, we will estimate the value of outliers on the ground of the upper fence and lower fence of the boxplot, which are calculated as follow:



$$upper\ fence = 75^{th}\ quantile + 1.5 * IQR$$
$$lower\ fence = 25^{th}\ quantile - 1.5 * IQR$$

Where $IQR = 75^{th}\ quantile - 25^{th}\ quantile$. According to those formulas, the summary of outliers for `balance` and `age` is presented in the right side.

	balance	age
lower_fence	-2004.000000	10.500000
upper_fence	3740.000000	70.500000
outlier_cnt	2943.000000	319.000000
outlier_percent	0.100146	0.010855

(After interpolate value 0)

3.1.4 Data pre-processing

- **Fill missing values**
  - `balance`

All 0 in `balance` have been filled with method. interplot().

  - `job` / `education`

Of these 2 variables, there are not exceed 5% of missing values, so the missing term will be filled with the mode of each variable.

- `contact`

`contact` is the enumerated type of data, which 2 values – cellular and telephone (excluding unknown). The detail of values frequency of `contact` is shown in right.

	frequency	fre_percent
cellular	19126	0.650832
unknown	8364	0.284616
telephone	1897	0.064552

In consideration that unknown takes almost 30% of total data, simply filling in the term with mode might impact the performance of algorithm modes, thus, we decide to randomly distribute `unknown` with cellular and telephone, based on the relative frequency between cellular and telephone, that is 9% (1897/ (19126+1897)) to telephone and 91% to cellular. We use the method random. random() from package random to randomly generate float between 0 and 1as the critical value to realise the distribution. The final result shows left also.

	frequency	fre_percent
cellular	26748	0.910198
telephone	2639	0.089802

- `poutcome`

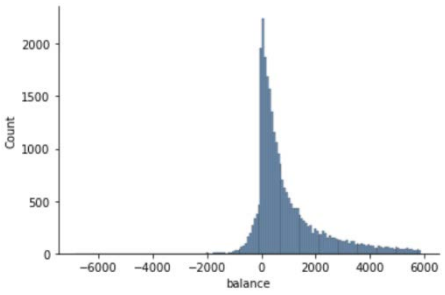
`poutcome` presents the outcome of the previous campaign. But, if the customer had not been contacted in the previous campaign, `poutcome` should logically miss. This can also be treated as a feature of the customer. thus, we will fill the `unknown` in `poutcome` variable with `not_contact_before`. There are still 4 unknown values since the above process is done. These will be allocated as `not_contact_before` either.

- **Delete outliers**
  - `age`

Ranged from 18 and 95. All outliers are reasonable; thus, we do not delete outliers for the variable.

- `balance`

There are 2943 outliers in balance. Of the outliers, 2933 are located above the upper fence, and 10 are located below the lower fence. We eventually decide the delete the head 5% of outliers above the upper fence. After the deletion, the distribution now looks more clearly.



3.2 Store

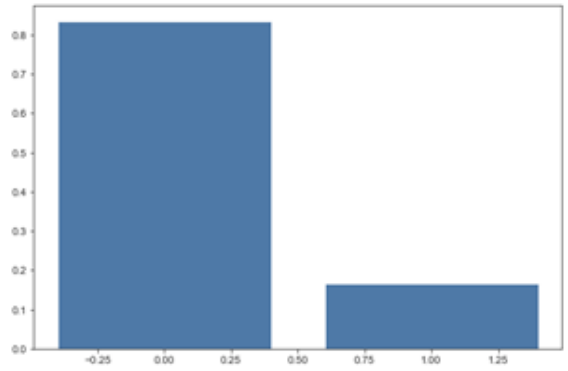
3.1.1 Data overview

The CSV file `store` given by the fashion store contains 21740 rows and 48 columns of data. Among the 48 data items, 47 are predictors and one is response value (`RESP`), In predictors, where `CC_ Card`, `VALPHON` and `WEB` are classified in `binary` types as the only type of qualitative data; Among all the remaining quantitative data, the data is roughly divided into `discrete` and `numerical_ percent` according to whether the data presentation result is a percentage or not. See the **Appendix 4** for specific divisions.

- **Binary features**

According to the histogram distribution of the `binary` data from **Appendix 5**, the identity characteristics of most customers are having a credit card, correctly filled in telephone information and online shopping users. Based on the displayed variable type, `VALPHON` data needs to digitize the variable name to 0&1 from N&Y, and this part will be carried out later.

- **Quantitative data and response variable**



As shown in **Appendix 6 and 7**, in the data exploration of quantitative data, it can be found that most of the data are right-skewed and the peak value is 0. This means that in many items of the datasets, the participation of customers in these items are low. The final `RESP` variable also supports the correctness of the above results. Overall, only about 18% of them will give some feedback on the merchant's activities which the above plot is shown.

As shown in **Appendix 8&9**, only the data of some projects show a strong positive correlation trend with the response variable (such as `CLASSES` & `COUPLES`), and a few data show a strong negative correlation (such as `AVRG`, `FREDAYS`). Other weak positive and negative correlation data have little impact on the response variables, which will not be specially emphasized in this part but will be used as prediction parameters in subsequent modelling.

3.2.2 Missing value

Through the missing value analysis of the discussed dataset, there is no obvious missing value need to amend or supply replacement values.

3.2.3 Outliers

Due to the discussed dataset without any continuous data category, there is no need to analysis whether the outliers in the dataset.

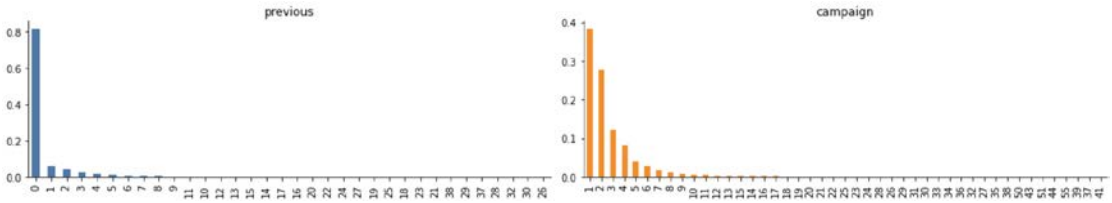
4. Feature Engineering

According to Predo Domingos *(Domingos, 2012)*, feature engineering is the most important process of a machine learning project. The purpose of feature engineering is to transform data into features that can express the essence of problems better so that the application of these features to the prediction models can improve the accuracy of the prediction of invisible data.

4.1 Bank

4.1.1 Sparse Categories

In categorical data, there might be some cases that a variable has a huge number of features but only occur in a few frequencies. One way to settle the problem is to consider that variable as continuous data. Otherwise, treating those features as a single variable might cause cures of dimension. Another way to dealing the problem is to group those rare features into one feature, called others. This method is sparse categories.



In the training dataset, `previous` and `campaign` has over 25 features, but most all those futures only occur 1 or 2 times. Our first solution is to find a boundary, and all values that exceed the boundary will be grouped into a single feature. However, we found that treating these values as continuous data can make our model get a higher accuracy for prediction. Thus, we eventually decide to treat these 2 variables as continuous data.

4.1.2 Feature transformation

For nominal data, we design dummy variables for each feature. In the end, we decide to treat all variables that obtain numerical values as numerical; For continuous data, we apply standardization to scale the value to a desirable range, thus the loss function can converge faster. Most features were scaled well, but `previous`, `campaign` and `pdays` are still right-skewed. The distribution of continuous data is presented in **Appendix 10**.

4.1.3 Mutual information

	MI
poutcome	0.0280
pdays	0.0258
month	0.0233
balance	0.0186
contact	0.0130
previous	0.0124
age	0.0108
housing	0.0093
job	0.0085
day	0.0077
campaign	0.0055
education	0.0026
loan	0.0023
marital	0.0017
default	0.0002

Mutual information presents the correlation between two variables and the extent to which of how they correlated. It is a broader special case of relative entropy, that is if the variables are not independent, then we can determine whether they are `close` to each other by examining the Kullback-Leibler divergence between the product of the joint probability distribution and the marginal probability distribution. The left table presents the mutual information between every feature we are going to use and the response variables. In conclusion, `poutcome` is the most correlated term with response variables, followed by `pdays`, `month` and `balance` etc.

4.1.4 Final decision for feature engineering

After the preprocessing and feature engineering, we used the processed dataset to train the models and found it did not achieve a satisfactory result. So, we decided to simplify the complicated preprocessing steps such as filling missing values and deleting outliers, and leave it to the tree-based models because tree-based models have characteristics that can handle these problems naturally.

We still kept the feature engineering process, including dummy categorical variables, discrete variables and the process of standardization, in order to get a better score of modelling.

4.2 Store

4.2.1 Data transformation

As mentioned earlier, `VALPHON` data needs to be converted for subsequent model operation. Before modification, the output results of the original data are Y and N, which respectively represent whether the customer has filled in the correct telephone information. After the replacement of the data, Y & N will be transformed into 1 & 0 in turn. **Appendix 11** is the change of data items before and after conversion.

4.2.2 Dataset classification

In order to facilitate the subsequent model establishment and performance test for the established model, using the `train_test_split` function divides the original data set into the training set and test set in a ratio of 7:3. After separation, the data volumes of the two are 15217 rows and 6523 rows respectively.

5. Modelling

A variety of algorithms with different capabilities and purposes have been proposed to accomplish this classification task, including Logistic Regression, Random Forest (RF), Gradient Boosted decision trees (GBDT), Light Gradient Boosting Machine (LGBM) and Model stacking. This paper experimented with these five machine learning models to make the best prediction for the response variable, `subscribed`. Each model would estimate the probability of the binary results. They then made the final binary prediction based on the probability threshold.

Logistic regression (or logit model) is a typical model for binary classification. It is based on a linear model for the log of odds (or logit of probability) and the principle of maximum likelihood *(Medium, 2021).* Among all five models, Logistic regression has the simplest nature, so it was selected to be the baseline model for this study. Lasso and Ridge's regression are extensions of logistic regression with regularization. They were used together to analyze the feature importance.

However, both datasets had many problems that challenged linear models' assumptions and affected their performance accuracy. These problems include imbalanced response variables, heavily skewed features, missing value (`unknown`) and multicollinearity as described in the previous section.

Meanwhile, tree-based models have characteristics that make them ideal for dealing with these issues. Tree-based models are the ensembled Decision tree algorithm. They have characteristics such as robustness to outliers, natural handling of mixed data type and missing values, insensitivity to the monotone transformation of inputs. Moreover, the tree-based model automatically did feature selection, which also simplified the

preprocessing process. Based on all these characteristics, this paper adopted tree-based models as the main algorithms. More importantly, trees are the foundation of many powerful machine learning algorithms. Three powerful tree-based algorithms, Random Forest, GBDT and LGBM would be introduced in detail below.

5.1 Random Forest (RF)

5.1.1 Description of the Algorithm

Random Forest, which is a form of a tree-based ensemble approach. The fundamental idea is that instead of developing one strong learner (single decision tree with all features), by developing many weak learners (small decision trees with subsets of features), their integrated prediction will outperform the strong learner's prediction. By growing these small trees without pruning, RF further reduced the overall overfitting risk.

Moreover, the correlation between trees is critical because the higher the correlation between the trees, the higher the prediction variance or the risk of overfitting. RF accomplished feature diversity by generating various subsets of random features as training input for each decision node. By doing so, RF reduced the correlation by avoiding feature dominance at the top nodes. Finally, averaging the predictions of small trees to get the final output.

5.1.2 Reason for choosing the model

The approach used to build each small tree with subsets of random features tree is the same as that used to build a single Decision tree. As a result, RF still retains all of the advantages of the tree model as mentioned before. While it also addresses the issue of overfitting by lowering the correlation between trees through feature diversity. As a result, RF can produce better forecasts than a single Decision tree in practice. So, RF was a suitable candidate model for these two binary classification tasks. However, the good performance does not come with no cost, RF required more hyperparameters to be tuned as well as more running time compared to a single Decision Tree.

5.1.3 Hyperparameter tuning

Optuna was used to tune the following hyperparameters and the optimal hyperparameters were listed below *("sklearn.ensemble.RandomForestClassifier", 2021).* Overfitting is a danger of hyperparameter optimisation. During the hyperparameter tuning process, 5-fold cross validation was employed to reduce the danger of overfitting.

1) n_estimators: the number of decision trees under RF. More trees in the forest, lower the bias. The increase of overfitting risk is slow because of the low correlation between individual trees in RF. "n_estimators" = 333 was optimal.
2) Criterion: impurity measurement in each node. "entropy" was found to be optimal for RF.
3) max_features: the feature size of the subset at each decision node is determined

by the split variable. In a grid search, the smaller the split variable, the less connection between the candidate parameter number.

4) min_samples_leaf: the minimal number of observations/samples required at each leaf node is specified by min samples leaf. Overfitting may be reduced by increasing the min samples leaf variable, which reduces the number of splits, as having too many splits is the major source of overfitting. In grid search, the range was set to 1 to 19.

The default values of max features (sqrt(n features)), min samples leaf = 14, and max-leaf nodes = 4 were discovered during hyperparameter tinkering.

5.2 Gradient Boosted decision trees (GBDT)

5.2.1 Description of the Algorithm

Another Decision Tree ensemble method is gradient boosting. Despite the fact that Random Forest is also an ensemble method, the two algorithms operate in quite distinct ways. Gradient Boosting develops trees sequentially, one tree at a time. During each iteration, the algorithm updated the observation's weight based on the residual of the prior models to fit the decision tree *(Si, Zhang, etc., 2017)*. Thus, observations with large errors will be given more weight. So, it can reduce the training error gradually as it proceeds.

5.2.2 Reason for choosing the model

As a consequence of GBDT's boosting feature introduced above, Gradient Boosting can often achieve superior overall performance than other ensemble approaches, such as Random Forest and Bagging. It made GBDT the ideal candidate model after Random Forest because we can further explore the performance difference resulting from different ensemble methods. In addition, it also obtained all the beneficial characteristics of the tree-based model described before. The ability to handle missing data, mixed data type and outliers are important to these two datasets. However, GBDT's high performance also comes with the cost of longer running time and the risk of overfitting, which will be addressed later in the hyperparameter tuning section.

5.2.3 Hyperparameter tuning

Even though GBDT generally performs better than RF because of its more effective boosting feature. It is inevitably exposed to the risk of overfitting. By tuning the regularization parameters below *("sklearn.ensemble.GradientBoostingClassifier", 2021)*, GBDT can effectively restrain the degree of overfitting while still maintaining its prediction performance. The hyperparameter tuning method was the same as RF, which is Optuna with 5-fold cross-validation.

1) n_estimators: it is the number of boosting iterations that GBDT will perform. Similar to RF, the overfitting risk increases slowly with increasing iteration. So, people generally select high values for this parameter. It was found the optimal value was 610 for the bank dataset and 689 for the fashion store dataset.

2) max_depth: it determines the depth of individual trees. As max_depth increases, deeper each tree will grow, increasing the likelihood of overfitting. It cannot be too small because the model may underfit. The optimal value was 3 for the bank dataset and 5 for the fashion store dataset.
3) learning_rate: it is a shrinkage parameter that intuitively represents the step size of the algorithm when it progresses toward the best prediction. Small learning_rate often give better prediction performance with the cost of much longer running time. The optimal value was found to be approximately 0.0825 for the bank dataset and 0.01360 for the fashion store dataset.

5.3 Light Gradient Boosting Machine (LGBM)

5.3.1 Description of the Algorithm

Light GBM is a more advanced application of gradient boosting compared to GBDT. It used a histogram-based algorithm *(LightGBM, 2021)* to significantly improve the training speed and memory usage efficiency. Moreover, LGBM divides the tree leaf by leaf with the best fit *(Khandelwal, 2021)*. In contrast to other boosting algorithms that split the tree by level. By doing so, Light GBM can reduce more loss and achieve higher levels of accuracy than any existing boosting algorithms.

5.3.2 Reason for choosing the model

As this report has thoroughly explained previously, GBDT is an efficient, accurate and widely-used machine learning algorithm. However, GBDT is facing challenges such as trade-off between efficiency (running time) and accuracy, especially for datasets contains with high dimensionality, the algorithm takes much longer time to achieve high accuracy. LGBT is able to achieve faster the training process when compared to traditional GBDT, while maintaining almost the same degree of performance. This feature made LGBT a great candidate model to be experimented.

5.3.3 Hyperparameter tuning

Although LGBM is a fast and high-performed model, it is still necessary to tune the hyperparameters to achieve the optimal configuration for the best evaluation, especially with the regularization parameters. The following is the parameter-tuning process and final adjusted outcomes.

1) ‘max_depth’: defines the maximum depth of the tree which can prevent overfitting.
2) ‘n_estimator’: a number of boosted trees to fit and as the more estimator can lead to better performance.
3) ‘num_leaves’: defines the number of leaves in the tree which should be less than 2*maximum depth to prevent overfitting. In this case, it is bigger than the threshold which could expose to overfitting which is between 1 and 50.

4) ‘lambda_l1’ and ‘lambda_l2’ are the regularization term of the weight which are positive and less than 1 and used to prevent overfitting.
5) ‘bagging_fraction’ and ‘bagging_freq’ are the percentage of data in each iteration and the frequency of bagging respectively. They are both used to increase the speed of the training process and prevent underfitting. Moreover, ‘min_data_in_leaf’ is the amount of data in each leaf. By setting this value which is relatively low can prevent the tree grow too deep and underfitting.

Finally, Model Stacking was used to combine the different model's output and use a logistic model as the meta-classifier which assigns the weight based on accuracy but did not perform expected results.

Optimal parameters	Bank	Fashion store
max_depth	3	2
n_estimator	670	603
num_leaves	46	9
lambda_l1	0.3066	0.0001
lambda_l2	0.0001	0.1045
bagging_fraction	0.9729	0.9510
bagging_freq	2	6

Optimal parameter values

6. Result Discussion

6.1 Bank

Since the bank dataset was given a test set on Kaggle, so it was used directly to evaluate the performance of each model. The metric on Kaggle was the AUC score which accounts for the class imbalance issue. According to the performance table below, it was clear that GBDT performed the best among all five models by achieving the highest AUC score of 0.80641 on the test set. As a result, GBDT will be chosen as our final model for the bank dataset.

The logistic regression did not perform well compared to other models because the data problems challenged its assumptions. However, it together with lasso and ridge regression, all provided useful interpretation for insight analysis. Random Forest performed well because it effectively reduced the overfitting problems through hyperparameter tuning. It was interesting to find Light GBM performed worse than both GBDT and RF. It may be because of the limited choices of hyperparameter range.

Models	Kaggle Score
Logistic Regression	0.77479
Random Forest	0.80297
Model Stacking	0.67747
GBDT	0.80641
Light GBM	0.795

Kaggle Scores of models for bank dataset

6.2 Store

6.2.1 Metric selection

For the Fashion Store dataset, we additionally design a loss matrix to compute the “tau” value for modifying the decision boundary. This study aimed to identify the positive cases (those customers who are more likely to respond to the fashion store’s email). So,

the prediction models need to minimize the false negative cases because the client (The fashion store) does not want to miss potential customers. Therefore, metrics targeting false-negative cases, such as sensitivity, false-negative rate, and F1 score have been selected to be our main metrics in model selection. Other general metrics such as accuracy would also be reviewed to gain a more complete picture of the model performance.

By modifying the decision boundary, several metrics could easily reach high scores. In other words, they are simply a measure of the model's performance with a given decision boundary. As a result, they aren't the most accurate predictor of the model's performance. ROC, on the other hand, can evaluate binary classification models without a decision boundary. It's also useful when dealing with the problem of unbalanced categorization. The AUC score (Area Under ROC) is a numerical representation of ROC evaluation that facilitates direct numerical comparison. As a result, the five models were also evaluated using the AUC score.

6.2.2 Result discussion

All performance metrics were listed in the table above. As this report mentioned before, the main evaluation metrics are False Negative rate and AUC scores. GBDT has the highest AUC score and F1 score as well as the second-lowest False Negative rate. Thus, it was chosen as the optimal model for the Fashion Store dataset.

	Accuracy	Sensitivity	False Negative Rate	F1 Score	AUC
Logistic	0.7077	0.7050	0.2918	0.4447	0.7813
Random Forest	0.7544	0.7368	0.2421	0.4991	0.8301
Model Stacking	0.8045	0.4391	0.1227	0.4272	0.6788
GBDT	0.7760	0.6981	0.2085	0.5086	0.8432
Light GBM	0.7585	0.7285	0.2355	0.5005	0.8385

Evaluation Matrix of Fashion Store dataset

Model Stacking has the lowest False Negative rate. Interestingly, model stacking also has the lowest AUC score but the highest Accuracy value. It confirmed the reason for choosing AUC as the main metric in the “Metric selection” section. Accuracy is just a metric of the final prediction result, which can be easily manipulated by the decision boundary of our chosen loss matrix. Whereas the AUC score measures the performance of the model via the probability without the decision boundary. The imbalanced response variable also contributed to the abnormally high accuracy and low AUC score of models stacking because accuracy failed to account for class imbalance. Therefore, AUC gave us a more unbiased performance estimate.

7. Limitation and future improvement

Although the overall performance for both datasets is satisfied, there are some

shortcomings of our methods and models. First, both datasets are highly imbalanced even after transformation. If datasets are more balanced which could lead to better performance. Second, we used dummy encoding for nominal features which could be replaced by other more effective encoding methods. Then, when hyperparameters are tuned, some of the models only use 20 trials with a timeout. If we increase the number of trails, we can find better configurations and better performance. For model stacking, the classifiers we use might not be the best combination which the hyperparameters of the underlying models also need to be tuned. Additionally, the parameter number of Light GBM may not be enough and the range of the parameters could be calculated and researched more efficiently.

8. Types of customers who are more responsive to marketing campaigns

8.1 Bank

8.1.1 Data insight of bank

As the Logistic coefficient plot of Figure 1 from **Appendix 12** shows in the exploration of Estimated Coefficients, the successful outcome of the previous marketing campaign (poutcome_success) shows the highest prediction power on the response variable "subscribe". Successful marketing campaigns produce successful outcomes. This phenomenon is also verified in the diagram of Feature Importance, where poutcome_success is the most important factor of all. This makes perfect sense due to that successful marketing campaigns have high probabilities of receiving success outcomes. Unknown contact communication types(contact_unknown) showed the strongest negative coefficient. This is also shown in Lasso coefficient Plot (Figure 2) and Ridge Coefficient Plot (Figure 3). Therefore, it is recommended to use cellular and telephone to contact consumers in marketing to improve the success rate of activities. And both Logistic and Lasso models indicate that March (month_mar) is the peak period for consumers to subscribe to bank time deposit services in a year.

Additionally, these can also be observed according to GBDT's chart (Figure 4). It is verified that the age level of consumers is a factor that cannot be ignored by sorting the graph according to the importance of features. Different age groups have different degrees of acceptance of marketing activities, and successful marketing activities should consider the audience of all age groups.

8.1.2 Suggestion for bank

1) Promotion activities through cellular and telephone
   Through the data analysis of the customer data set, it is known that the unknown contact method has a negative impact that cannot be ignored on the customer's willingness to subscribe to the bank's fixed deposit service. Therefore, banks should try to avoid this phenomenon. Instead, contacting customers through cellular and telephone as much as possible for marketing can help increase the success rate of subscriptions.
2) Customize different marketing strategies according to different age groups
   The influence of age as a positive correlation should also be considered. Customer

groups of different ages have different acceptance of marketing activities. It is necessary that to divide customer groups according to age groups and customize different marketing strategies. If the marketing activities are more targeted, it can also help increase the success rate of the activities.

3) Increase publicity in March every year
   The coefficient of "month contact" in March is the most prominent, indicating that customers are most willing to accept subscription services in March. Banks can consider March every year as a good opportunity to develop new customers. Holding more marketing activities in March will make it easier to attract new customers than holding activities to promote subscription services in other months.

8.2 Store

8.2.1 Data insight of store

Likewise, the prediction results of Logistic, Lasso and Ridge models for the data set of stores are also shown in the figures from **Appendix 13**. The Logistic model shows the four factors most likely to cause customers to respond negatively to marketing activities. The smaller the value of the Lifetime Average of days between Visits (LTFREDAY), the fewer times customers spend browsing products. This is the data that most intuitively reflects customers' degree of interest in products since this feature has the strongest negative coefficient. The average amount spent per visit (AVRG) and the number of days between purchases (FREDAYS) rank second and third in negative correlation. Consumption amount and interval days are very important indicators to analyze consumer purchasing power. Customers with low purchasing intention are also less likely to respond to marketing activities. Finally, if the product consistency (HI) is not high, consumers will not be attracted.

Both the Lasso and Ridge models (figure 2&3) reflect that number of purchase visits (FRE) is an important factor for referring to whether consumers respond to marketing activities. People with more purchases are stickier to the brand and more willing to respond to advertising emails. Due to the web shopper (WEB) is highly dependent on the Internet, they are the group that accepts marketing advertising emails most, hence they also present the highest positive correlation coefficient in the Lasso model. Similarly, the negative coefficients of LTFREDAY and PC_CALC20 are also very high in these two models.

8.2.2 Suggestion for store

1) Focus on customer needs and place advertisements more accurately
   The number of purchases, lifetime average of days between visits, and advertising consistency reflect the customer's consumption preferences. Fashion stores should place different advertisements according to consumers' consumption preferences. Ensure that the marketing advertisements are consistent with the types of products they have consumed in the past. It can help increase consumers' attention to advertising, and they will also respond more actively.
2) Advertise more for target customers

It is worth noting that customer groups with high consumption amounts and short time intervals for each consumption have a high degree of stickiness to the brand. They have both the willingness to consume and the ability to consume. This group of major customers who contribute spending credits to the store, the fashion store can position these customers as target customers and increase marketing efforts to increase the effectiveness of marketing.

3) Find potential customers

Groups with high values of "Lifetime average of days between visits" have shown a strong interest in the client's fashion store. This type of group can be screened for a second time and divided into two types of people with consumption records and those without consumption records. Those who have no record of consumption but spend a lot of time browsing advertisements are potential customers of the fashion store. Intensifying marketing efforts for them can convert some of them into actual customers. This can also effectively improve the success rate of marketing activities.

9. Conclusion

In this report, we compute 5 models which are Logistic Regression, Random Forest, Model Stacking, GBDT, and Light GBM for our clients to predict whether a customer will subscribe to a term deposit and whether a customer will respond to a promotion email. Among all models, GBDT has the best performance for both datasets which have AUC scores of 0.8064 and 0.8432. GBDT has effectively predicted the most important features for the bank and fashion store. Outcome and customer age of successful marketing activities are the most important factors that banks need to consider when holding marketing activities. And Lifetime Average of days between visits is the most important variable for the fashion store. Clients need to consider these factors when conducting marketing campaigns to help them market effectively.

Reference List

Domingos, P. (2012). A few useful things to know about machine learning. Communications Of The ACM, 55(10), 78-87. https://doi.org/10.1145/2347736.2347755.

Khandelwal, P. (2017, 6 12). Which algorithm takes the crown: Light GBM vs XGBOOST? Retrieved 11 10, 2021, from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

LightGBM. (2021). Features. Retrieved 11 12, 2021, from LightGBM: https://lightgbm.readthedocs.io/en/latest/Features.html

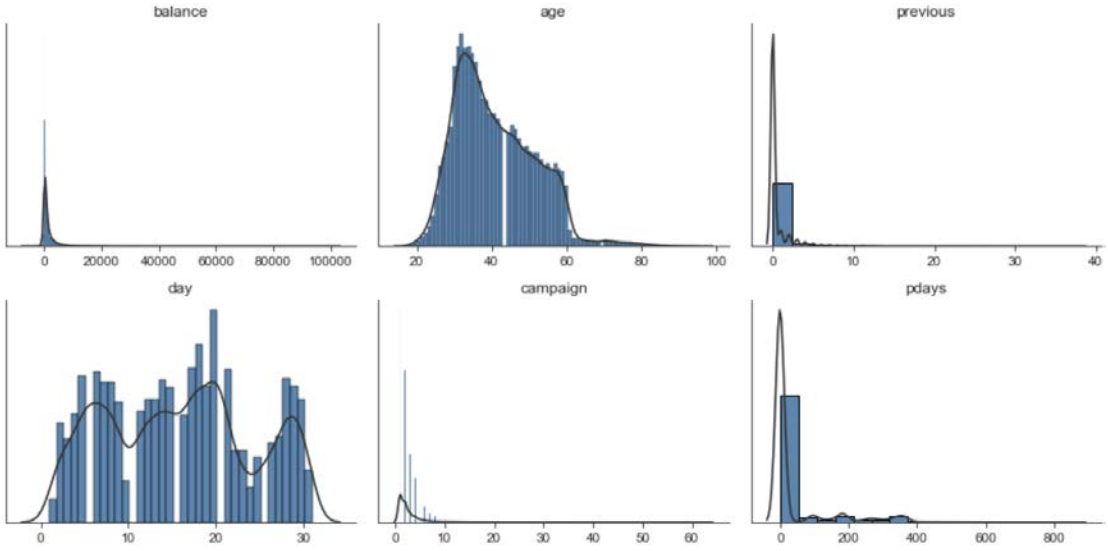
Logistic Regression — Detailed Overview. Medium. (2021). Retrieved 11 November 2021, from https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc.

Si, S., Zhang, H., Keerthi, S.S., Mahajan, D., Dhillon, I.S. & Hsieh, C.. (2017). Gradient Boosted Decision Trees for High Dimensional Sparse Output. Proceedings of the 34th International Conference on Machine Learning, in Proceedings of Machine Learning Research 70:3182-3190 Available from https://proceedings.mlr.press/v70/si17a.html

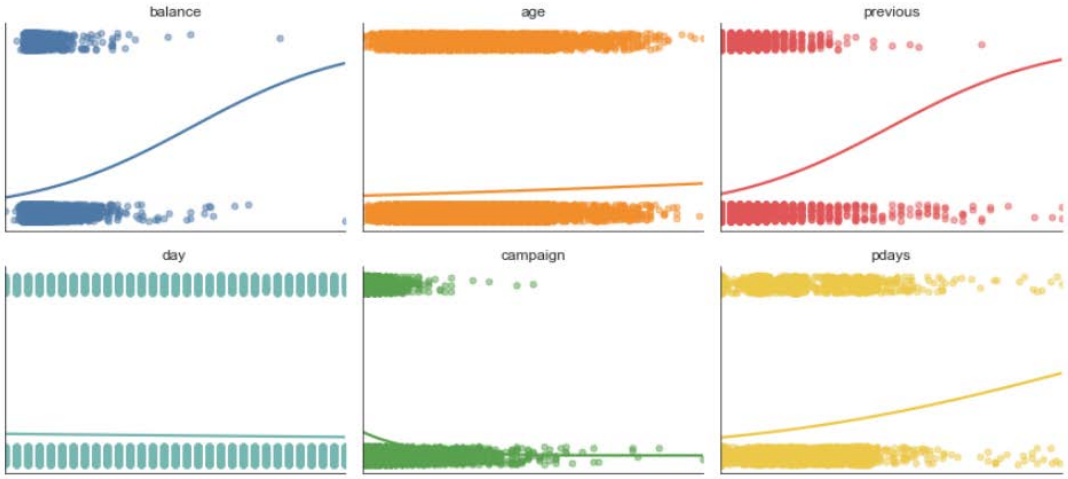
Sklearn.ensemble.RandomForestClassifier. scikit-learn. (2021). Retrieved 12 November 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

Appendix List:

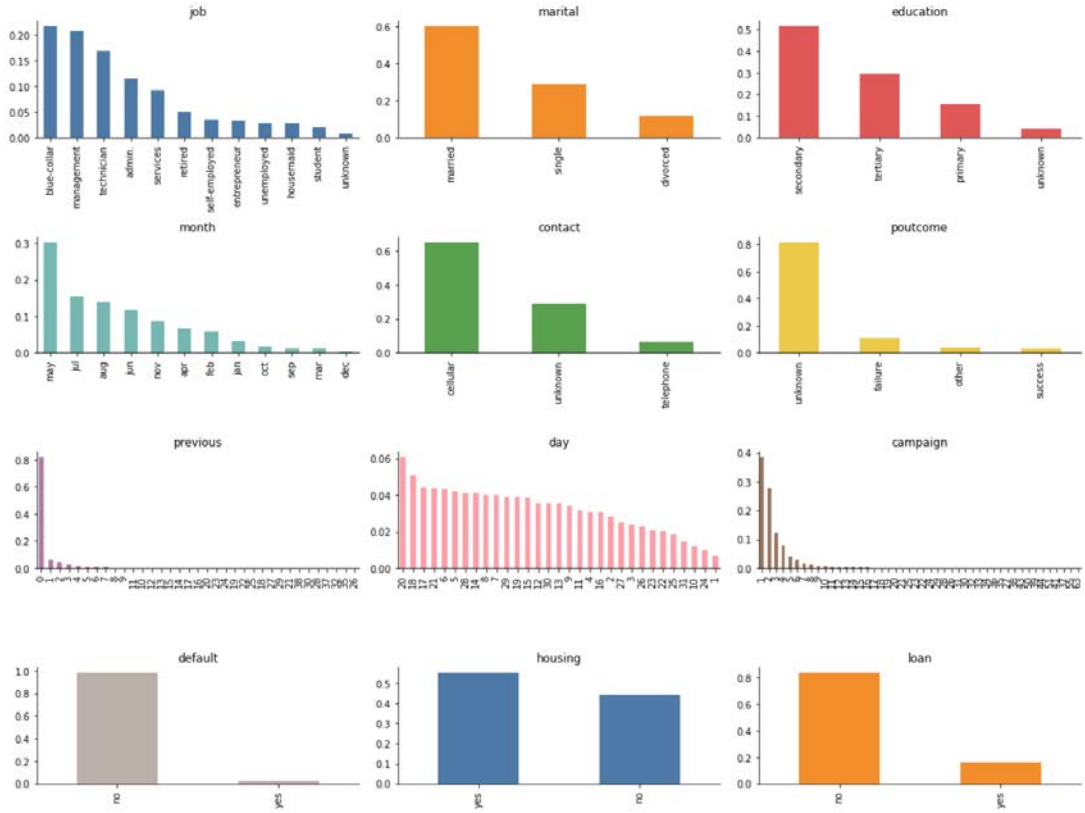
1. Quantitative variables – density distribution – histogram



2. Continuous features – Measure of dependence



3. Qualitative features:

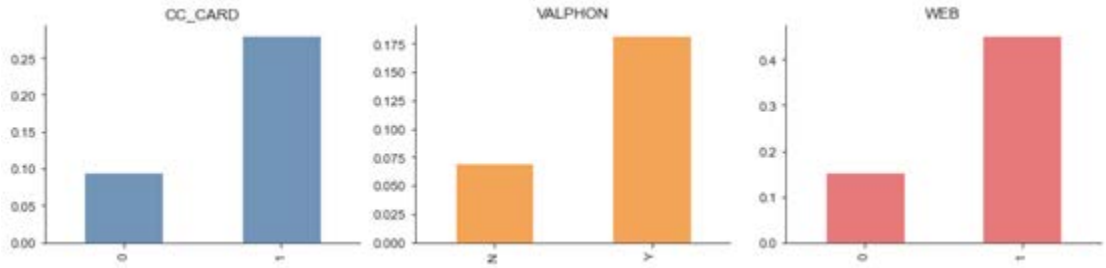


4. Quantitative variables – store

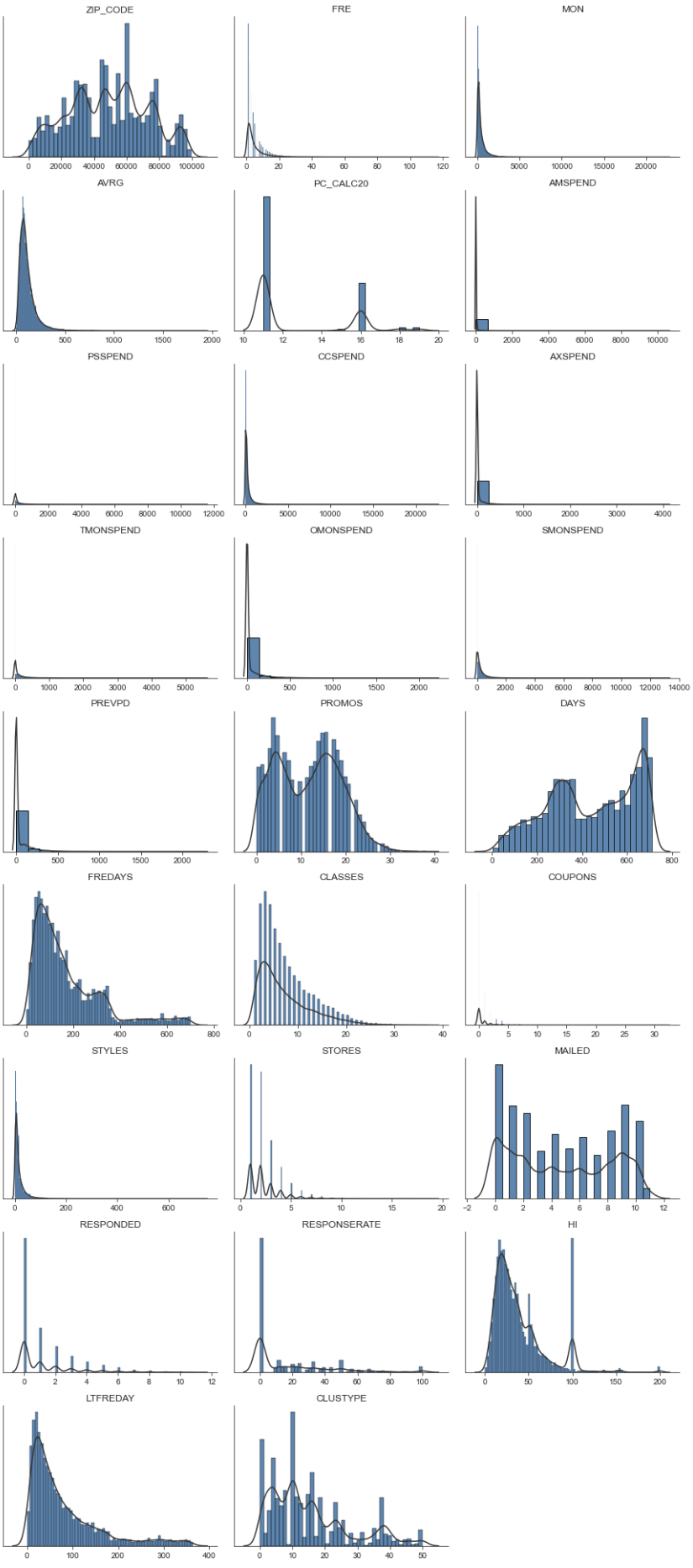
```
# quantitative variables
discrete = ['ZIP_CODE', 'FRE', 'MON', 'AVRG', 'PC_CALC20','AMSPEND', 'PSSPEND',
            'CCSPEND', 'AXSPEND', 'TMONSPEND', 'OMONSPEND', 'SMONSPEND', 'PREVPD',
            'PROMOS', 'DAYS', 'FREDAYS','CLASSES', 'COUPONS', 'STYLES', 'STORES',
            'MAILED', 'RESPONDED', 'RESPONSERATE', 'HI', 'LTFREDAY', 'CLUSTYPE']

numerical_percent = ['PSWEATERS','PKNIT_TOPS', 'PKNIT_DRES', 'PBLOUSES', 'PJACKETS',
                     'PCAR_PNTS', 'PCAS_PNTS', 'PSHIRTS', 'PDRESSES', 'PSUITS', 'POUTERWEAR',
                     'PJEWELRY','PFASHION', 'PLEGWEAR', 'PCOLLSPND','GMP', 'MARKDOWN','PERCRET']
```

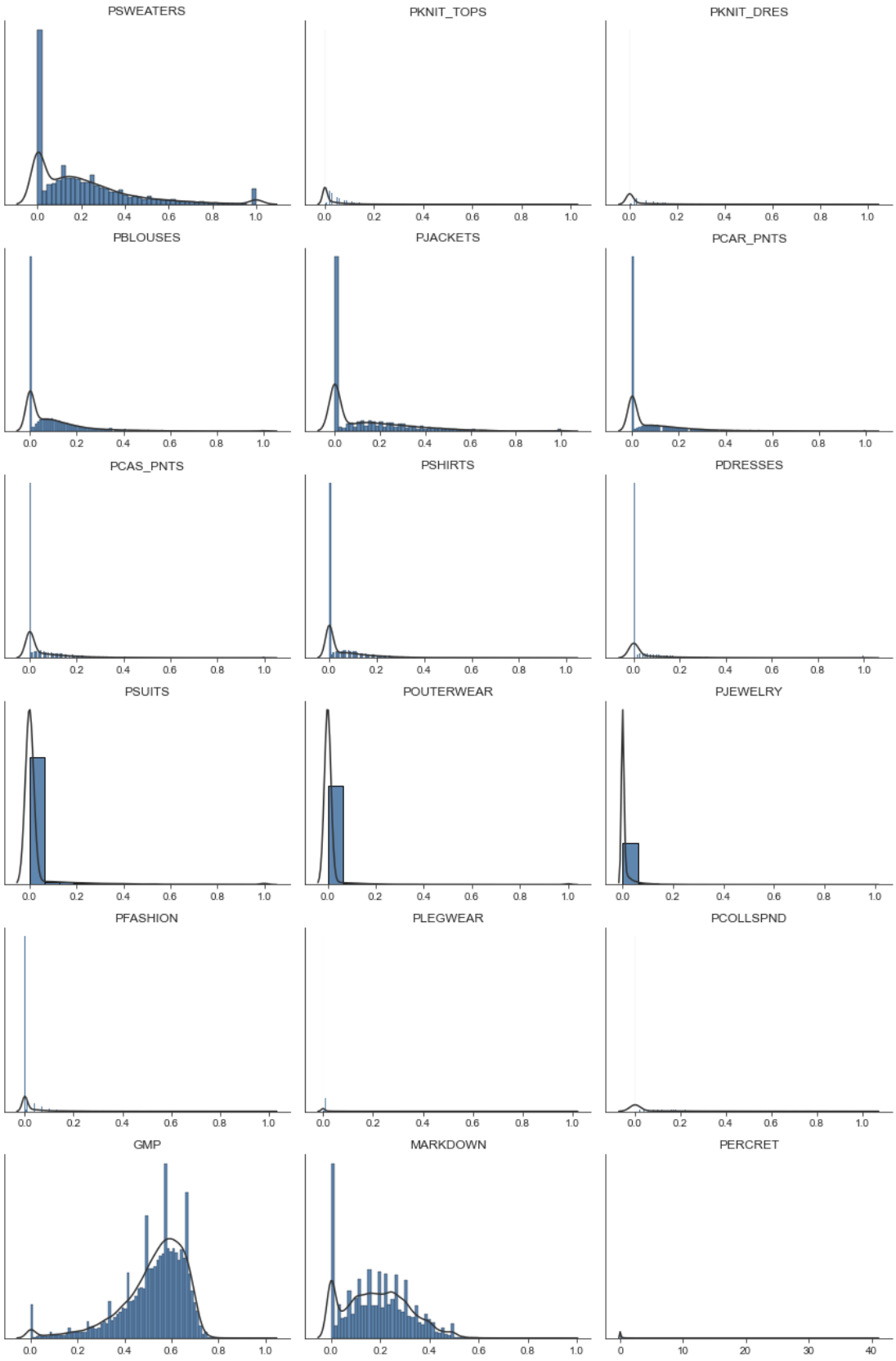
5. Binary variables – store



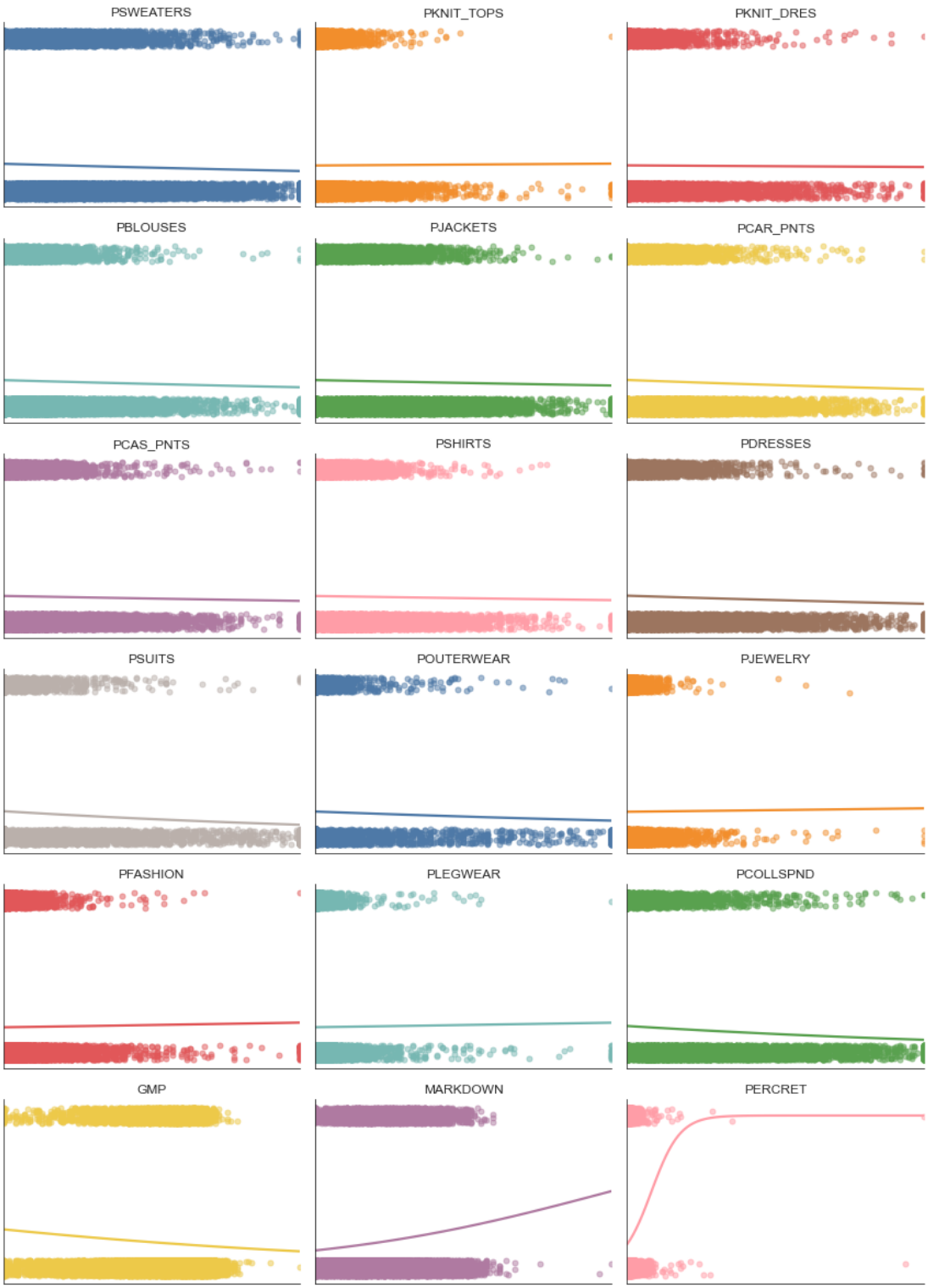
6. Discrete data – store – distribution


ZIP_CODE
FRE
MON
AVRG
PC_CALC20
AMSPEND
PSSPEND
CCSPEND
AXSPEND
TMONSPEND
OMONSPEND
SMONSPEND
PREVPD
PROMOS
DAYS
FREDAYS
CLASSES
COUPONS
STYLES
STORES
MAILED
RESPONDED
RESPONSERATE
HI
LTFREDAY
CLUSTYPE

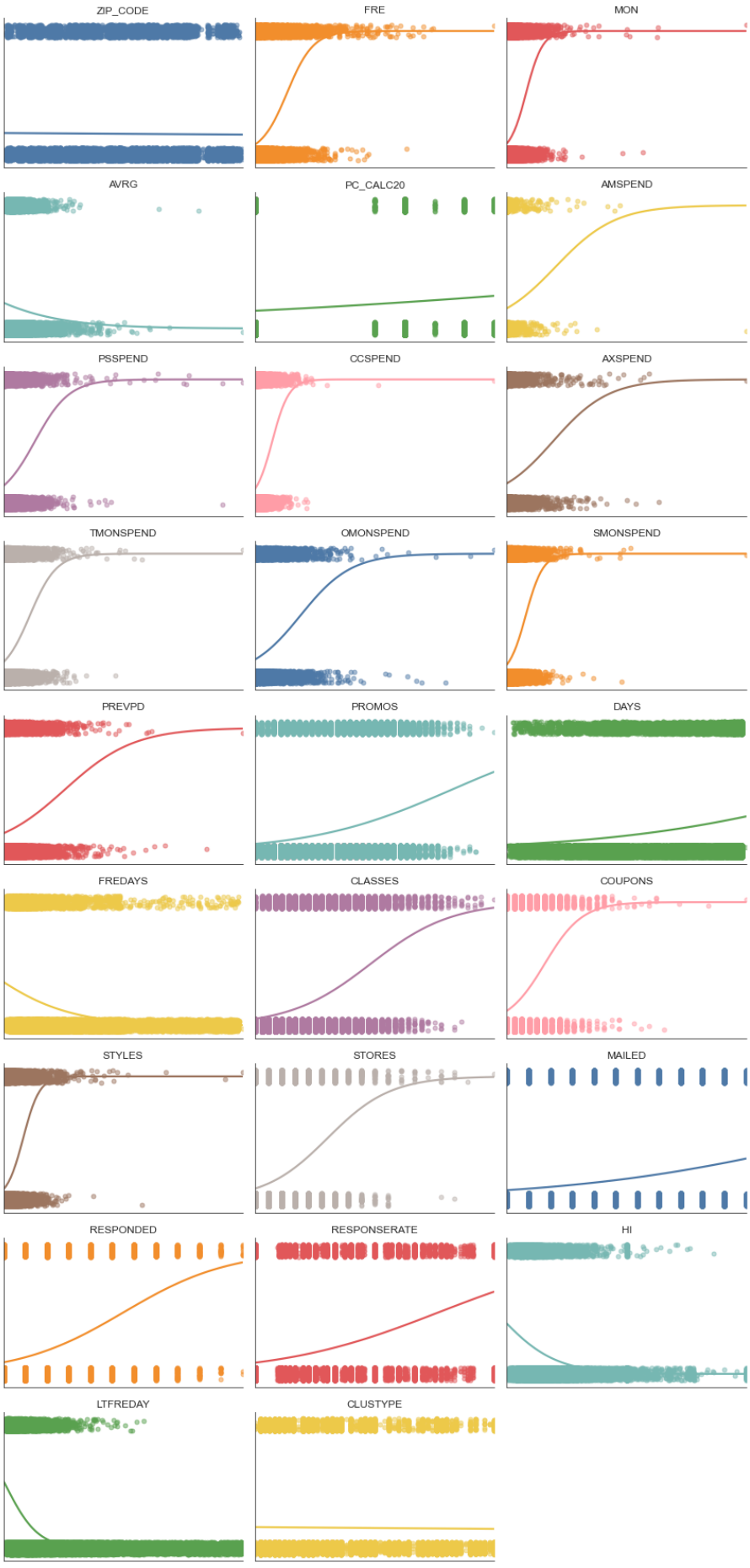
7. Numerical_percent data – store – distribution



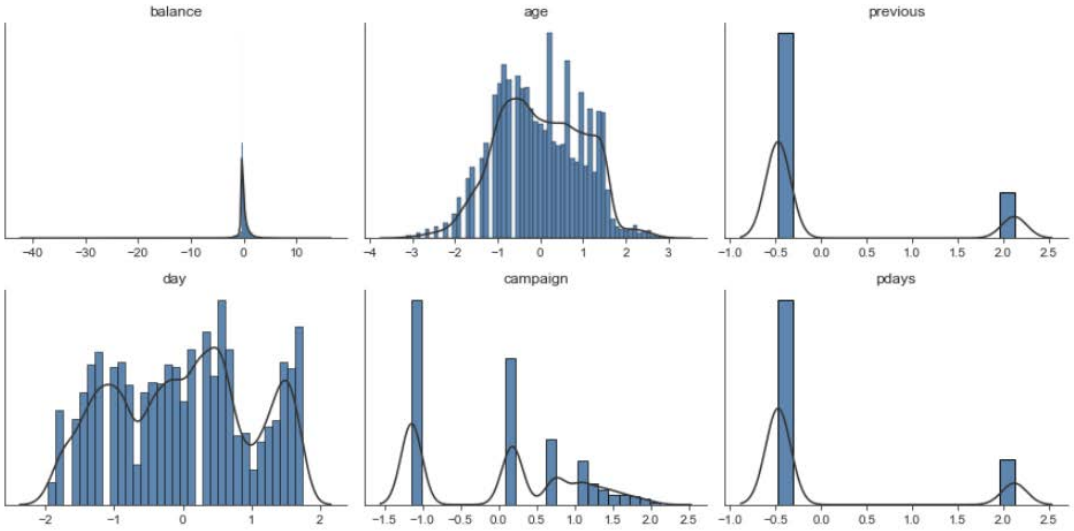
8. Discrete data – store – regression plot



9. Numerical_percent data – store – regression plot


ZIP_CODE
FRE
MON
AVRG
PC_CALC20
AMSPEND
PSSPEND
CCSPEND
AXSPEND
TMONSPEND
OMONSPEND
SMONSPEND
PREVPD
PROMOS
DAYS
FREDAYS
CLASSES
COUPONS
STYLES
STORES
MAILED
RESPONDED
RESPONSERATE
HI
LTFREDAY
CLUSTYPE

10. Distribution of continuous – Bank



11. `VALPHON` - variable information – before & after transformation

```
0        N
1        Y
2        N
3        Y
4        Y
        ..
21735    Y
21736    N
21737    N
21738    Y
21739    Y
Name: VALPHON, Length: 21740, dtype: object
```

```
0        0
1        1
2        0
3        1
4        1
        ..
21735    1
21736    0
21737    0
21738    1
21739    1
Name: VALPHON, Length: 21740, dtype: object
```

12. Data insight – bank

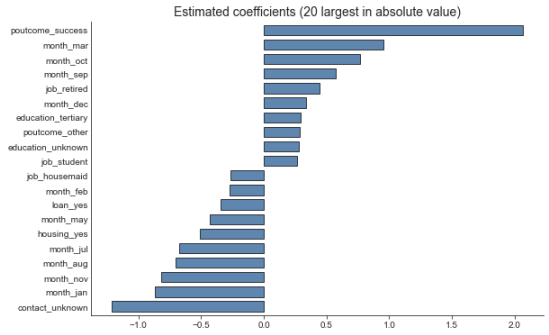


Figure 1. Logistic coefficient plot of Bank

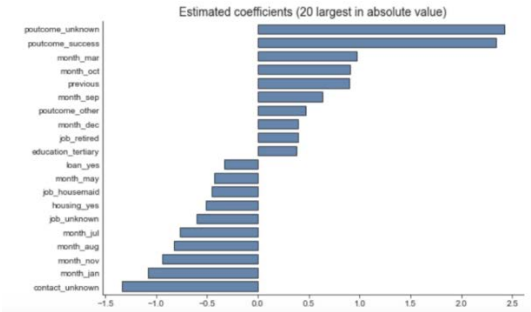


Figure 2. Lasso coefficient plot of Bank

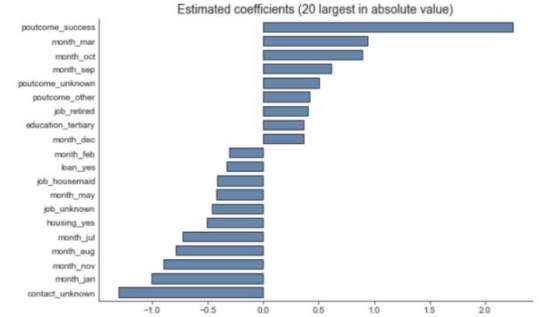


Figure 3. Ridge coefficient plot of Bank

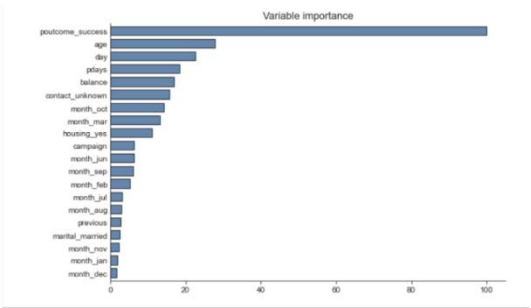


Figure 4. GBDT of bank

13. Data insight – store

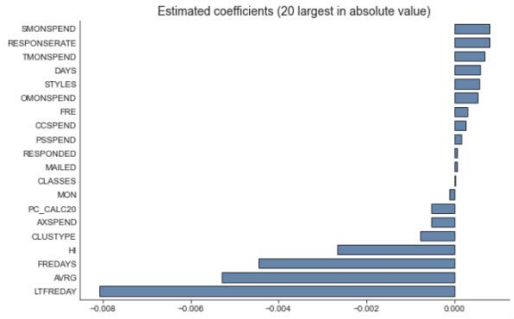


Figure 1. Logistic coefficient plot of Bank

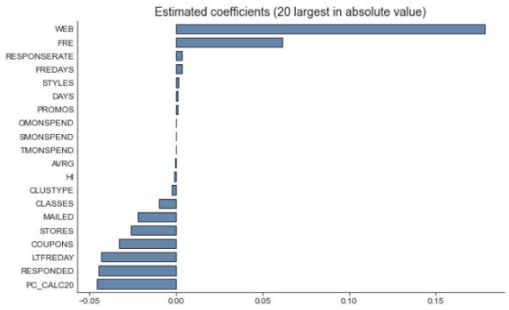


Figure 2. Lasso coefficient plot of Store

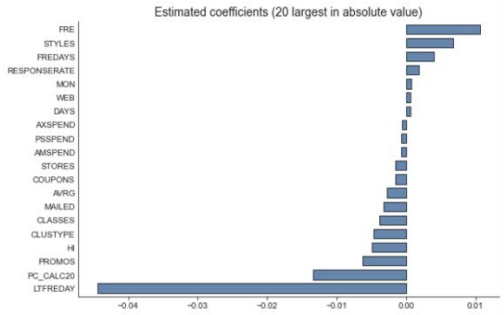


Figure 3. Ridge coefficient plot of Store