Lead Conversion Prediction System with Machine Learning Models

December 2023

Written by: Thao Phuong Vu, William Cheong, Aparna Sawant, Gengchen Song, Yuki Yu

I. Business Understanding:

Lead conversion is important in banking operation's success but sales and marketing activities are costly in time and money. World Plus, a mid-size private bank, want to deploy lead prediction system to identify potential customers who will convert and buy new term products to have strategies for those customers through communication channels. From World Plus' data set, we have two objectives in this report.

- **Objective 1 (Priority):** minimising cost by identifying potential customers accurately.
- **Objective 2:** increasing revenue by reaching as many potential customers as possible.

II. Data preparation:

First, we must ensure our dataset is free of errors, missing values, and anomalies that bias model prediction. Therefore, we perform a thorough data preparation process, where we will perform data cleaning, handling of missing values and outlier detection to ensure data's suitability for predictive models. We begin by checking errors such as duplicated customers and missing values. Despite no duplicated customer, 18268 missing values were detected on the credit product variable. We replaced missing values to the mode of data ("No") for that column to ensure we do not lose too much valuable information. Besides, an anomaly (-1) was found in "Dependent" variable with no meaning, thus, we removed it from dataset.

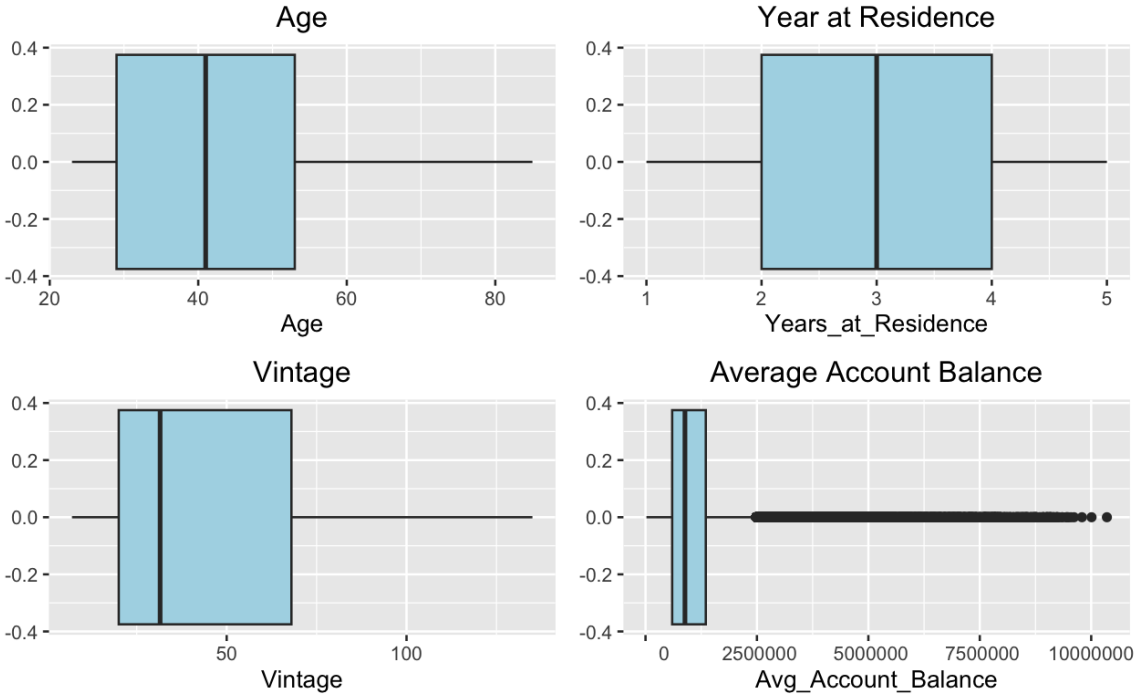


Figure 1: The box plot of Age, Year at Residence, Vintage and Avg_Account_Balance variables

We used boxplots to check the continuous variables' distribution (Avg_Account_Balance , Age, Vintage, Year_at_Residence). For "Avg_Account_Balance" variable (Figure 1), we observed some data with much higher values than variable's mean. However, we keep them to prevent a biased model as higher account balance customers is meaningful to target variable.

We used 'ifelse' function to update values' name in 'Gender', 'Active', and 'Credit_Product' variables, converting non-numeric values to numeric values '1' and '0'. Then, we applied one-hot encoding to encode categorical variables (Region_Code, Marital_Status, Occupation, Channel_Code, Account_Type), which contain non-numeric values that must be transformed into numeric representations for the model to understand and utilise them effectively.

```
> # Check the class distribution in the target column for data_lead,
trainingset and testset
> prop.table(table(data_lead$Target))

       No       Yes
0.8523071 0.1476929
> prop.table(table(training$Target))

       No       Yes
0.8523045 0.1476955
> prop.table(table(test$Target))

       No       Yes
0.8523134 0.1476866
```

Figure 2: The class distribution in the target column for data_lead, trainingset and testset

We partitioned whole dataset into training (70%) and test (30%). From Figure 2, the class imbalance with 85% no and 15% yes in target variable will negatively affect the model's accuracy and hamper its capacity to support decisions based on its output. Lastly, we removed ID column as it is not essential and may undermine the predictive model's quality.

Johnson and Khoshgoftaar (2020) used Medicare data with high class imbalance to compare the performances of oversampling (ROS), undersampling (RUS) and hybrid method (over and undersampling). Accordingly, hybrid method outperforms ROS and RUS in AUC score and G-Mean performance, indicating that hybrid method can effectively balance the class distribution and improve model's performance. They also mentioned that failure to handle imbalance may bias models toward the majority class and misclassify

instances belonging to the minority class more often than those in the majority class. The hybrid method offers flexibility, effectively balancing classes, preventing overfitting, and enhancing model's generalization. Thus, we choose the hybrid sampling to balance the data, ensuring a 50% distribution (Figure 3).

```
> # Check the class distribution in the target column for bothsampled
> prop.table(table(bothsampled$Target))

       No       Yes
0.5002989 0.4997011
```

Figure 3: The class distribution in the target column for 'bothsampled'

III. Modelling

We implemented four models to classify prospective bank customers: Support Vector Machine (SVM), Random Forest, Logistic Regression and eXtreme Gradient Boosting (XGB). The binary variable, 'Target', referred to lead conversion in given dataset and was used to analyse the models' results and determine their accuracy. Due to their unique advantages, these supervised models were selected to predict Target variable.

We implemented two distinct strategies for models: one with feature selection and one without. This approach is crucial because while feature selection based on information gain can mitigate overfitting, it might also miss important interactions between variables, potentially impacting model performance. Considering the operation efficiency and time cost, 10% of training data is used for parameter tuning, and K-fold cross-validation is set to 5 to select optimal model from these parameters. Once the best parameters are identified, we train model on the entire training set. Unlike basic models that directly classify customers, our model predicts the probability of a customer making a purchase, allowing for the manual setting of different threshold values for future model comparison. Finally, the model's performance is evaluated using a separate test set.

1. SVM

SVM proves to outperform traditional prediction models in diverse marketing prediction scenarios. Cui and Curry (2005) compared SVM with multinomial logit model and found SVM had better predictive performance, particularly when dealing with large-scale and high-dimensional datasets. During the data preparation phase, we obtained a training dataset containing 153,918 observations with 31 variables and found that some customers have notably high bank account balances. Given that SVM works by finding hyperplane that best separates data into different classes to maximise the margin between hyperplane and the

closest data points. This allows SVM to handle non-linear problems and be robust to outliers. Thus, SVM is believed to predict prospective bank customers well. In model tuning, cost and gamma are two main parameters that will affect the performance of SVM. To avoid overfitting and underfitting, we set (0.1, 1, 10, 100, 1000) and (0.005, 0.5, 50) as the value range of cost and gamma, using 5-fold cross-validation to identify parameter combinations that perform the best prediction on unseen data.

2. Random Forest (RF)

RF is a more advanced ensemble model for classification prediction which can work well with missing values and imbalanced datasets. Miguéis, Camanho and Borges (2017) demonstrated its high accuracy in predicting consumer's response to direct marketing in Portuguese banking industry. In RF, each tree is built from original data with randomly assigned features (mtry). At each node, a subset of the available features is randomly selected and optimally segmented for that node (nodesize). The final prediction probabilities result from averaging individual tree predictions, allowing RF to reduce outliers' effect and overfitting compared with Decision Tree. In model tuning, we adjusted values of mtry and nodesize to find optimal parameter combinations, continuing until the improvement in AUC was less than 0.001.

3. Logistic Regression (LR)

Although LR has limitations in handling nonlinearity and multicollinearity, it is still an excellent candidate model. Firstly, it provides high computational efficiency, making it suitable for the complex data sets of this project. Additionally, it provides easy-to-interpret results and coefficients, which is useful when explaining model to non-technical people.

We apply Lasso regularization, following Tibshirani's 1996 work, in training LR models. Lasso's main advantage is minimising the residual sum of squares while imposing an upper bound on the absolute sum of coefficients, thus facilitating feature selection. This method is particularly suitable for many features' data sets, especially when some features have little impact on target variable. Furthermore, Lasso regularization offers stability for datasets with highly correlated features. Hence, we chose Lasso regularization over information gain due to the dataset's complexity.

4. XGB

XGB, a decision tree-based ensemble machine learning method, utilise gradient boosting to combine multiple low-performance models into a highly accurate composite model to minimise the loss function. It is known for delivering precise predictions by aggregating outcomes from various individual trees (Noorunnahar, Chowdhury & Mila, 2023).

Lee et al. (2021) identified that among the eight selected machine learning algorithms, XGB stands out for its superior ability in predicting online consumers' purchasing behaviour. This is because XGB mitigates overfitting through pre-pruning and post-pruning techniques during tree growth. Furthermore, they also enhance XGB interpretability by combining it with Explainable Artificial Intelligence (XAI) methods. This allows global assessment of each feature importance and detailed analysis of each feature's specific contribution to predicted outcomes for individual cases. This clear explanation makes the opaque XGB model more accessible to non-technical individuals, fostering broader trust in decision-making process. Thus, we use XGB and will utilize proposed XAI methods to enhance XGB's interpretability.

IV. Evaluation:

1. Deciding threshold

By using confusion matrix to evaluate performance, determining threshold is important. Esposito et al.(2021) found that 0.5 default classification threshold is often not ideal for imbalanced data. Adjusting decision threshold proved to be an effective strategy to deal with class imbalance in real-world projects. However, they also emphasised that threshold determination must be based on problem objective and the desired balance between false positives and false negatives. Therefore, we will change threshold to match our objectives.

Dowling (2023) argued that banking companies spent more than 20% of their overall budgets on advertising while banking's lead conversion rate is relatively low at only about 4.3-5% and about 15% for top banks in 2022-2023. Moreover, Kirby (2023) predicts that deposit rate will continue to decrease as individual and corporate customers seek better interest rates and households struggle with higher costs, resulting in lower likelihood of customers purchasing new term deposit products. Therefore, to reduce unnecessary costs, we set a high threshold of 0.8 and target customers with an 80% probability of buying new term deposit product.

2. Evaluating results

		SVM	Random Forest (RF)	Logistic Regression	XGB
Threshold 0.8	Accuracy	0.8770	0.9006	0.8988	0.9058
	Precision	0.5793	0.7581	0.7231	0.7551
	Recall	0.6097	0.4799	0.5096	0.5361
	F1	0.5941	0.5878	0.5979	0.6271

		SVM	Random Forest (RF)	Logistic Regression	XGBoost
Threshold 0.5	Accuracy	0.8231	0.8646	0.8214	0.8334
	Precision	0.4403	0.5337	0.4348	0.4601
	Recall	0.7298	0.6600	0.6972	0.7367
	F1	0.5492	0.5902	0.5356	0.5664

Table 4: Table of result of 4 models (with Feature Selection) (Refer to Appendix 3)

We chose a result with Feature Selection to minimise overfitting and improve accuracy. From Table 4, compared to threshold 0.5, there is a significant increase in the accuracy and precision results at threshold 0.8 as we decrease positive predictions while increasing negative predictions. RF and XGB provide great result with over 90% accuracy and over 75% precision, showing that among our predicted customers, more than 75% will purchase new term deposits.

		XGB	Random Forest (RF)	Difference between XGB & Random Forest	% Difference between XGB & Random Forest
Threshold 0.8	Accuracy	0.9058	0.9006	-0.0052	-0.52%
	Precision	0.7551	0.7581	0.0030	0.30%
	Recall	0.5361	0.4799	-0.0562	-5.62%
	F1	0.6271	0.5878	-0.0393	-3.93%

Table 5: Table of comparison the result between XGB and Random Forest

From Table 5, due to Precision's insignificant difference, we focus on model that can optimize revenue. XGB's Recall is higher, meaning it can accurately predict 54% of real potential customers. Furthermore, XGB's higher F1, demonstrating a better balance between precision and recall compared to RF.

However, Wynants et al. (2019) demonstrated high thresholds' risks by presenting multiple risk thresholds' results in estimating a disease risk's patient. They found that high threshold

makes model overly conservative (reduce recall and increase false negatives), leading to missed positive instances (real patients) and reduced overall performance. Besides, it reduces model's sensitivity in underlying data distribution or patterns, resulting in model obsolescence or ineffectiveness over time. In our case with threshold 0.8, we have to trade off the possibility of missing out actual potential customers, who have less than 80% likelihood to deposit new term, affecting revenue. Therefore, besides precision and recall, we explore additional metrics to evaluate our model.

3. ROC & AUC

Employing AUC to evaluate customer churn, Burez and Poel (2009) investigated performance improvement of sampling and modelling techniques over some standard methods. They proved that AUC provides a comprehensive evaluation for imbalanced classification because it reflects prediction probability ranking independently of classification threshold. This paper's findings are consistent with our imbalance data and uneven costs of false positives and false negatives. Thus, we choose AUC to determine our best model.

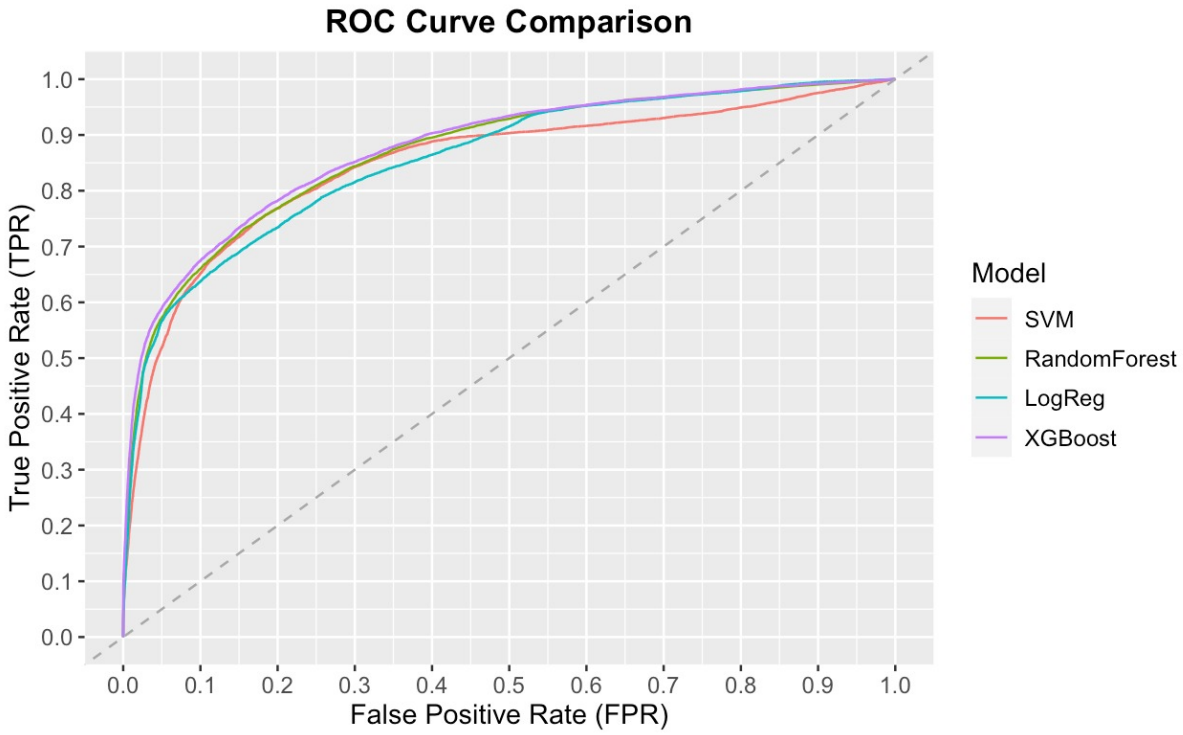


Figure 6: Receiver Operator Characteristic (ROC) Curve Comparision for four models

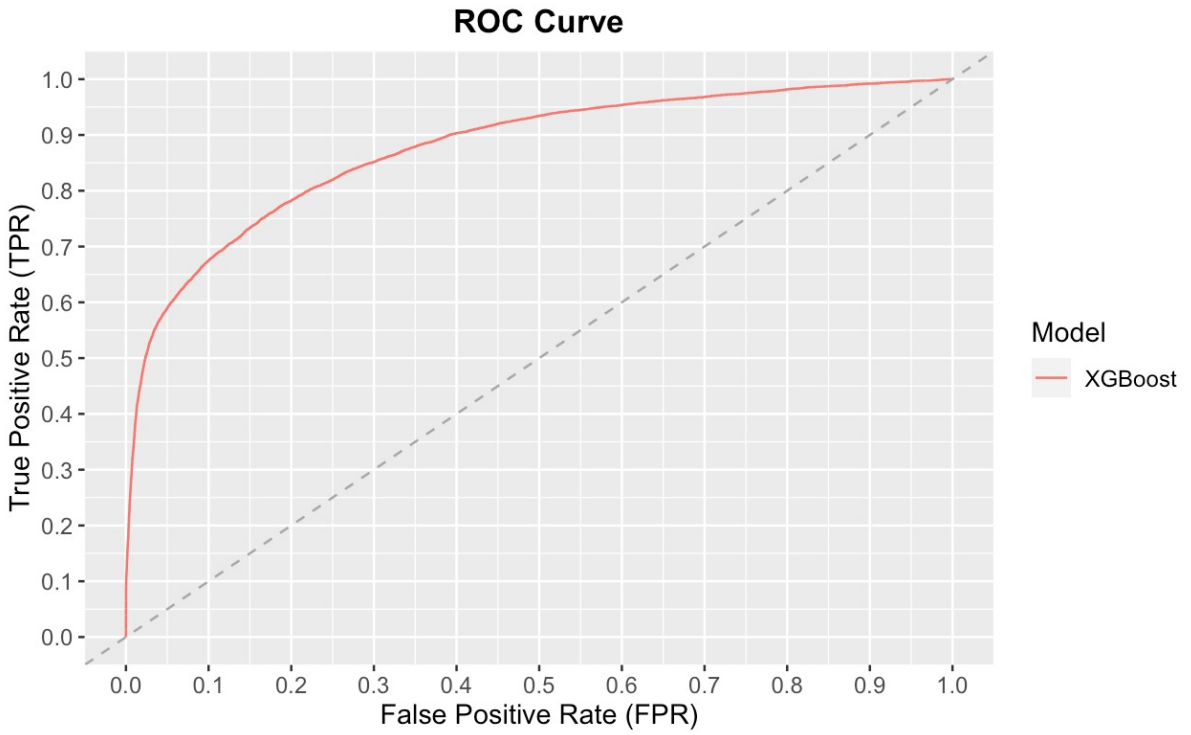


Figure 7: Receiver Operator Characteristic (ROC) Curve of XGB

	SVM		Random Forest		Logistic Regression	XGB	
	Without FS	With FS	Without FS	With FS		Without FS	With FS
AUC	0.8340	0.8481	0.8714	0.8694	0.8561	0.8751	**0.8765**

Note: *FS is Feature Selection*

Table 8: AUC Result of four models in both case with and without Feature selection.

From Figure 6, compared to other models, XGB's ROC Curve is closest to the top-left corner, indicating higher sensitivity (or TPR) and less False Positive mistakes across various threshold setting. This shows that XGB has the best performance and the highest accuracy. Furthermore, from Table 8, XGB's AUC with FS (0.8765) is nearest to 1.0, indicating that it covers the most area under ROC curve with a strong discriminative ability and is the best model at correctly classifying between customers who make purchases and who do not.

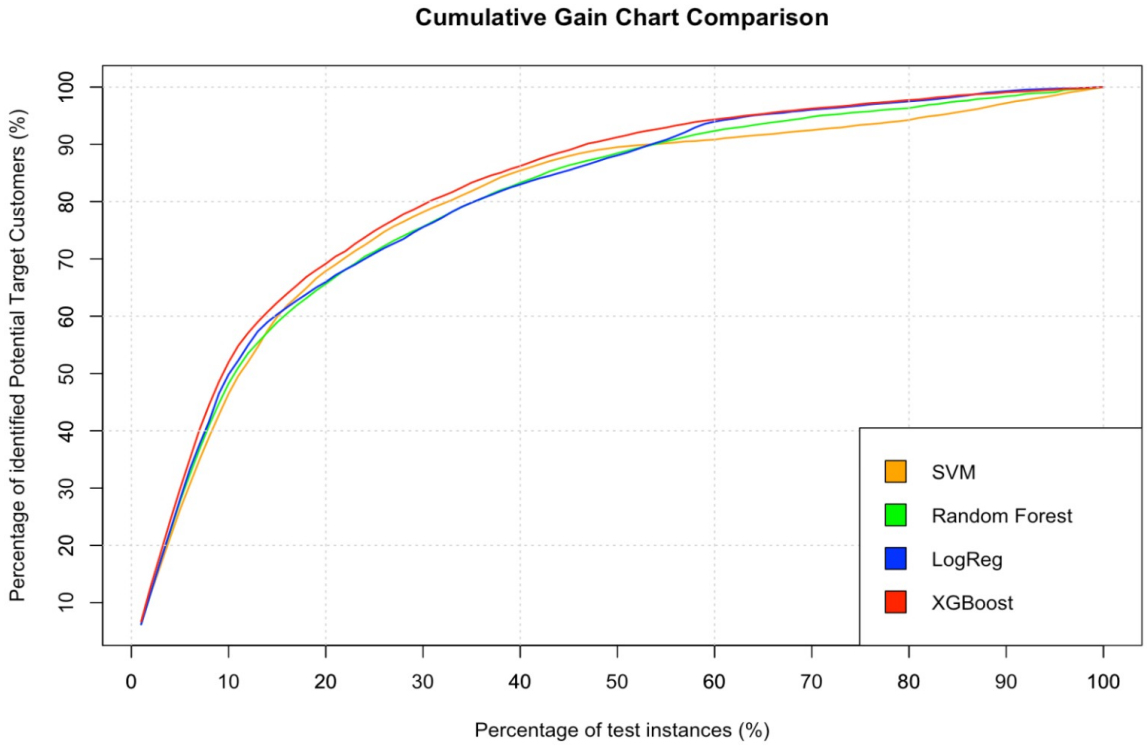


Figure 9: Cumulative Gain Chart Comparison for four models

XGB's gain curve is the steepest among other models' curve, demonstrating XGB is more efficient in prioritising instances with a higher likelihood of being positive, which is crucial in our case of identifying accurately potential customers to minimise unnecessary costs. Specifically, given same percentage of target customers, XGBoost can capture the highest percentage of customers who actually buy.

V. Deployment and Conclusion:

In conclusion, XGB, our chosen model, excels in handling large and complex datasets, delivering accurate predictions while mitigating overfitting. However, there are still potential limitations during the deployment stage.

Firstly, despite enhanced interpretability with XAI tools, it remains more intricate than simpler models like decision trees, posing a problem when explaining model decisions is necessary. Secondly, while XGBoost performs well with large-scale datasets, its prediction speed may lag behind simpler models, crucial in applications requiring rapid responses. Finally, the model's use of numerous trees (ntrees=200) demands significant computational resources. This could become a potential issue if World Plus Bank has limited computational resources.

VI. References:

Burez, J., & Poel, d. V. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 4626-4236.

Cui, D., & Curry, D. (2015). Prediction in Marketing Using the Support Vector Machine. *MARKETING SCIENCE*, 595-615.

Dowling, L. (2023, June 14). *Pathmonk*. Retrieved from Pathmonk: https://pathmonk.com/financial-industry-guide-conversion-rate-optimization/#:~:text=Banking%3A%20The%20average%20conversion%20rate,from%201%25%20to%203%25

Esposito, C., Landrum, G. A., Schneider, N., Stiefl, N., & Riniker, S. (2021). GHOST: Adjusting the Decision Tzhreshold to Handle Imbalanced Data in Machine Learning. *JOURNAL OF CHEMICAL INFORMATION AND MODELING*, 2623-2640.

Johnson, J. M., & Khoshgoftaar, T. M. (2020). The Effects of Data Sampling with Deep Learning and Highly Imbalanced Big Data. *Information Systems Frontiers*, 1113-1131.

Kirby, J. (2023, November 30). *THE TIMES money mentor*. Retrieved from https://www.thetimes.co.uk/money-mentor/mortgage-property/when-will-interest-rates-go-down-uk#:~:text=Analysts%20have%20said%20with%20inflation,by%20the%20end%20of%202025

Lee, J., Jung, O., Lee, Y., Kim, O., & Park, C. (2021). A Comparison and Interpretation of Machine Learning Algorithm for the Prediction of Online Purchase Conversion. *Thoretical and Applied Electronic Commerce Research*, 1472-1491.

Miguéis, V. L., Camanho, A. S., & Borges, J. (2017). Predicting direct marketing response in banking comparison of class imbalance methods. *Springer*, 831-849.

Noorunnahar , M., Chowdhury, A. H., & Mila, F. A. (2023, March 27). *PLOS ONE.* Retrieved from PLOS ONE: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0283452

Tibshirani, R. (2018, December 05). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, pp. 267-288.

Wynants, L., Smeden, v. M., McLernon, J. D., Timmerman, D., Steyerberg, W. E., & Calster, V. B. (2019, October 25). *Open Access*. Retrieved from https://link.springer.com/article/10.1186/s12916-019-1425-3

VII. Appendix:

Appendix 1: Data Dictionary

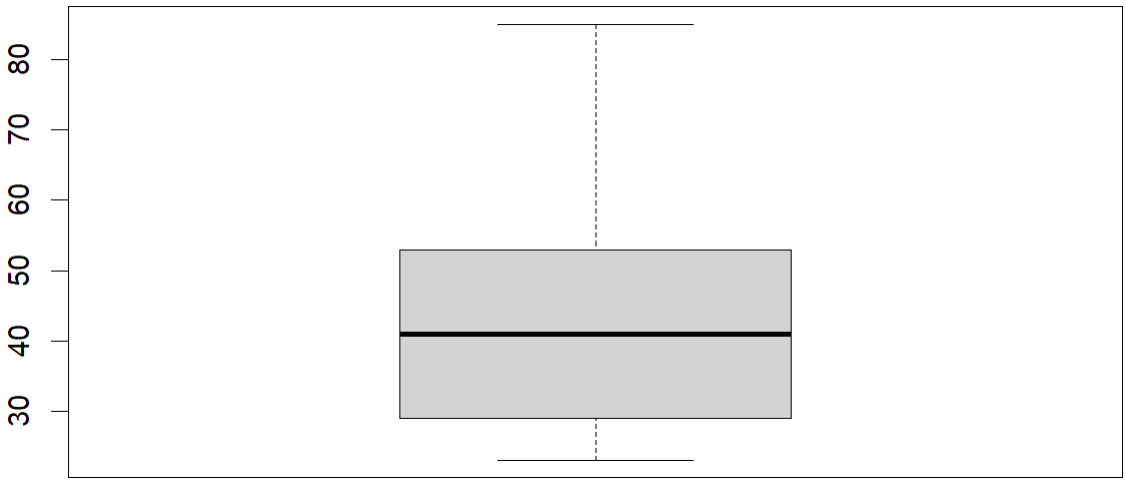
The details for the dataset provided by World Bank shows as below

1. Number of Instances: 220000
2. Number of Variables: 16
3. Attribution information

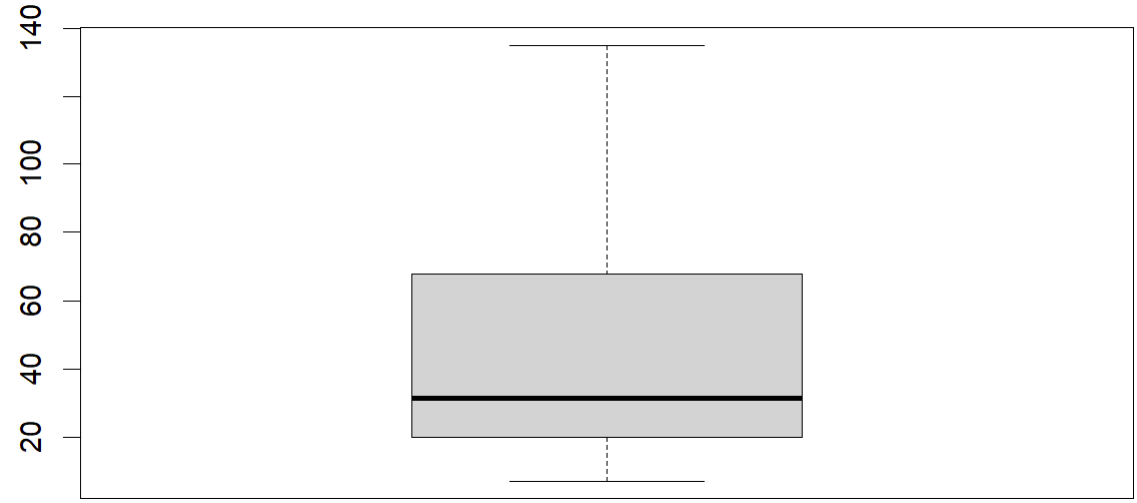
Name	Description
ID	Customer identification number
Gender	Gender of the customer
Age	Age of the customer in years
Dependent	Whether the customer has a dependent or not
Marital_Status	Marital state (1=married, 2=single, 0 = others)
Region_Code	Code of the region for the customer
Years_at_Residence	The duration in the current residence (in years)
Occupation	Occupation type of the customer
Channel_Code	Acquisition channel code used to reach the customer when they opened their bank account
Vintage	The number of months that the customer has been associated with the company.
Credit_Product	If the customer has any active credit product (home loan, personal loan, credit card etc.)
Avg_Account_Balance	Average account balance for the customer in last 12 months
Account_Type	Account type of the customer with categories Silver, Gold and Platinum
Active	If the customer is active in last 3 months
Registration	Whether the customer has visited the bank for the offered product registration (1 = yes; 0 = no)
Target	Whether the customer has purchased the product, (0: Customer did not purchase the product, 1: Customer purchased the product)

Appendix 2: Data Preparation

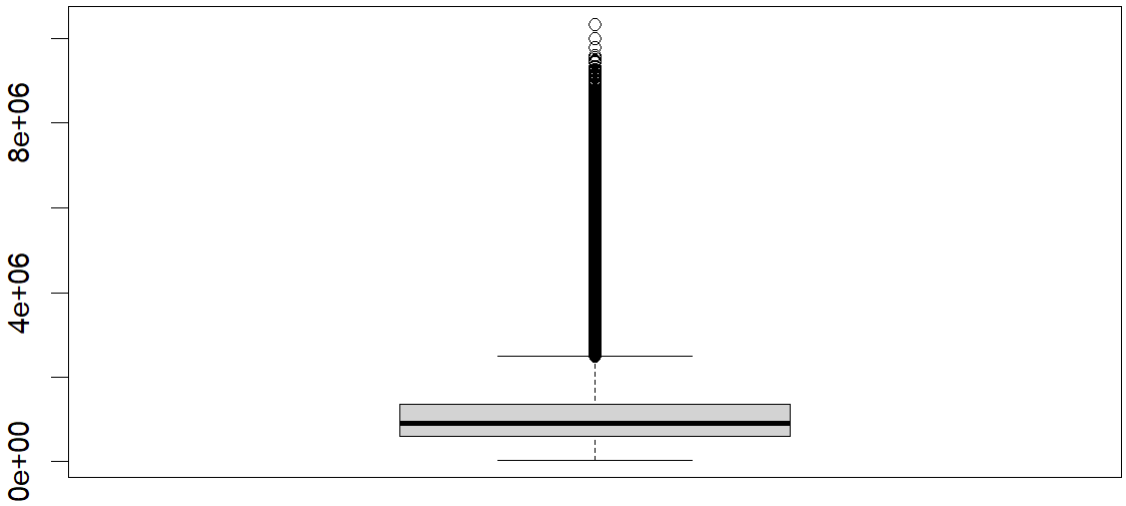
The box plot of Age variable



The box plot of Vintage variable



The box plot of Avg_Account_Balance variable



Appendix 3: Result of Models

Confusion Matrix and Model Result – SVM with feature selection (threshold 0.8)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  51909  3802
       Yes  4313  5940

               Accuracy : 0.877
                 95% CI : (0.8744, 0.8795)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5217

 Mcnemar's Test P-Value : 1.501e-08

              Precision : 0.57934
                 Recall : 0.60973
                     F1 : 0.59415
             Prevalence : 0.14769
         Detection Rate : 0.09005
   Detection Prevalence : 0.15543
      Balanced Accuracy : 0.76651

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – SVM with feature selection (threshold 0.5)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  47183  2632
       Yes  9039  7110

               Accuracy : 0.8231
                 95% CI : (0.8201, 0.826)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4474

 Mcnemar's Test P-Value : <2e-16

              Precision : 0.4403
                 Recall : 0.7298
                     F1 : 0.5492
             Prevalence : 0.1477
         Detection Rate : 0.1078
   Detection Prevalence : 0.2448
      Balanced Accuracy : 0.7845

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – SVM without feature selection (threshold 0.8)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  52420  4353
       Yes  3802  5389

               Accuracy : 0.8764
                 95% CI : (0.8738, 0.8789)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4972

 Mcnemar's Test P-Value : 1.126e-09

              Precision : 0.5863
                 Recall : 0.5532
                     F1 : 0.5693
             Prevalence : 0.1477
         Detection Rate : 0.0817
   Detection Prevalence : 0.1393
      Balanced Accuracy : 0.7428

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – SVM without feature selection (threshold 0.5)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  55103  5297
       Yes  1119  4445

               Accuracy : 0.9027
                 95% CI : (0.9004, 0.905)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5304

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.79889
                 Recall : 0.45627
                     F1 : 0.58082
             Prevalence : 0.14769
         Detection Rate : 0.06739
   Detection Prevalence : 0.08435
      Balanced Accuracy : 0.71818

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – Random Forest with feature selection (threshold 0.8)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  54730  5066
       Yes  1492  4676

               Accuracy : 0.9006
                 95% CI : (0.8983, 0.9029)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5345

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.75811
                 Recall : 0.47998
                     F1 : 0.58781
             Prevalence : 0.14769
         Detection Rate : 0.07089
   Detection Prevalence : 0.09351
      Balanced Accuracy : 0.72672

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – Random Forest with feature selection (threshold 0.5)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  55103  5297
       Yes  1119  4445

               Accuracy : 0.9027
                 95% CI : (0.9004, 0.905)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5304

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.79889
                 Recall : 0.45627
                     F1 : 0.58082
             Prevalence : 0.14769
         Detection Rate : 0.06739
   Detection Prevalence : 0.08435
      Balanced Accuracy : 0.71818

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – Random Forest without feature selection (threshold 0.8)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  55103  5297
       Yes  1119  4445

               Accuracy : 0.9027
                 95% CI : (0.9004, 0.905)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5304

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.79889
                 Recall : 0.45627
                     F1 : 0.58082
             Prevalence : 0.14769
         Detection Rate : 0.06739
   Detection Prevalence : 0.08435
      Balanced Accuracy : 0.71818

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – Random Forest without feature selection (threshold 0.5)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  52316  3718
       Yes  3906  6024

               Accuracy : 0.8844
                 95% CI : (0.882, 0.8869)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.5445

 Mcnemar's Test P-Value : 0.03222

              Precision : 0.60665
                 Recall : 0.61835
                     F1 : 0.61244
             Prevalence : 0.14769
         Detection Rate : 0.09132
   Detection Prevalence : 0.15054
      Balanced Accuracy : 0.77444

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – Logistic Regression (threshold 0.8)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  54321  4777
       Yes  1901  4965

               Accuracy : 0.8988
                 95% CI : (0.8964, 0.9011)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.542

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.72313
                 Recall : 0.50965
                     F1 : 0.59790
             Prevalence : 0.14769
         Detection Rate : 0.07527
   Detection Prevalence : 0.10409
      Balanced Accuracy : 0.73792

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – Logistic Regression (threshold 0.5)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  47393  2950
       Yes  8829  6792

               Accuracy : 0.8214
                 95% CI : (0.8185, 0.8243)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4323

 Mcnemar's Test P-Value : <2e-16

              Precision : 0.4348
                 Recall : 0.6972
                     F1 : 0.5356
             Prevalence : 0.1477
         Detection Rate : 0.1030
   Detection Prevalence : 0.2368
      Balanced Accuracy : 0.7701

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – XGBoost with feature selection (threshold 0.8)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  54528  4519
       Yes  1694  5223

               Accuracy : 0.9058
                 95% CI : (0.9036, 0.908)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5749

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.75510
                 Recall : 0.53613
                     F1 : 0.62705
             Prevalence : 0.14769
         Detection Rate : 0.07918
   Detection Prevalence : 0.10486
      Balanced Accuracy : 0.75300

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – XGBoost with feature selection (threshold 0.5)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  47799  2565
       Yes  8423  7177

               Accuracy : 0.8334
                 95% CI : (0.8306, 0.8363)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4701

 Mcnemar's Test P-Value : <2e-16

              Precision : 0.4601
                 Recall : 0.7367
                     F1 : 0.5664
             Prevalence : 0.1477
         Detection Rate : 0.1088
   Detection Prevalence : 0.2365
      Balanced Accuracy : 0.7934

       'Positive' Class : Yes

Setting levels: control = No, case = Yes
Setting direction: controls < cases
Area under the curve: 0.8765
```

Confusion Matrix and Model Result – XGBoost without feature selection (threshold 0.8)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  54532  4536
       Yes  1690  5206

               Accuracy : 0.9056
                 95% CI : (0.9034, 0.9078)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5736

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.75493
                 Recall : 0.53439
                     F1 : 0.62580
             Prevalence : 0.14769
         Detection Rate : 0.07892
   Detection Prevalence : 0.10454
      Balanced Accuracy : 0.75216

       'Positive' Class : Yes
```

Confusion Matrix and Model Result – XGBoost without feature selection (threshold 0.5)

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  47981  2641
       Yes  8241  7101

               Accuracy : 0.835
                 95% CI : (0.8322, 0.8379)
    No Information Rate : 0.8523
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4705

 Mcnemar's Test P-Value : <2e-16

              Precision : 0.4628
                 Recall : 0.7289
                     F1 : 0.5662
             Prevalence : 0.1477
         Detection Rate : 0.1076
   Detection Prevalence : 0.2326
      Balanced Accuracy : 0.7912

       'Positive' Class : Yes
```