STA 207 Project 4, Bank Marketing

Team ID: 12

Name (responsibilities): Joseph Gonzalez (Proofread, Introduction, Background)

Name (responsibilities): Yanhao Jin (Logistic Regression, Random Forest, Model Comparison)

Name (responsibilities): Ruichen Xu (Descriptive analysis)

Name (responsibilities): Bohao Zou (Logistic Regression Diagnostics)

1. Introduction

1.1 Background

Businesses rely on data-driven solutions to overcome economic instability and contend with new competitors. These data-driven solutions often reflect customer characteristics and employ data-mining techniques to analyze or predict customer behavior. Using the information obtained from the customers' data, businesses can strategically plan initiatives to influence attention to their services, maintain their current customer base, and expand their reach to new clients. From 2008 to 2013, a Portuguese retail bank conducted a direct marketing campaign to persuade new customers to commit to a long term deposit with favorable interest rates. The bank communicated with customers through telephone calls independently and, during these calls, they documented the customers' personal characteristics and whether they said "yes" of "no" to signing up for a long term deposit. In this project, we are interested in building classification models to predict whether a customer will commit to a long-term deposit. We establish logistic regression model, random forest model and xgboost model for the bank market data, evaluate the performance of the fitted model and then compare all the fitted models.

1.2 Descriptive analysis

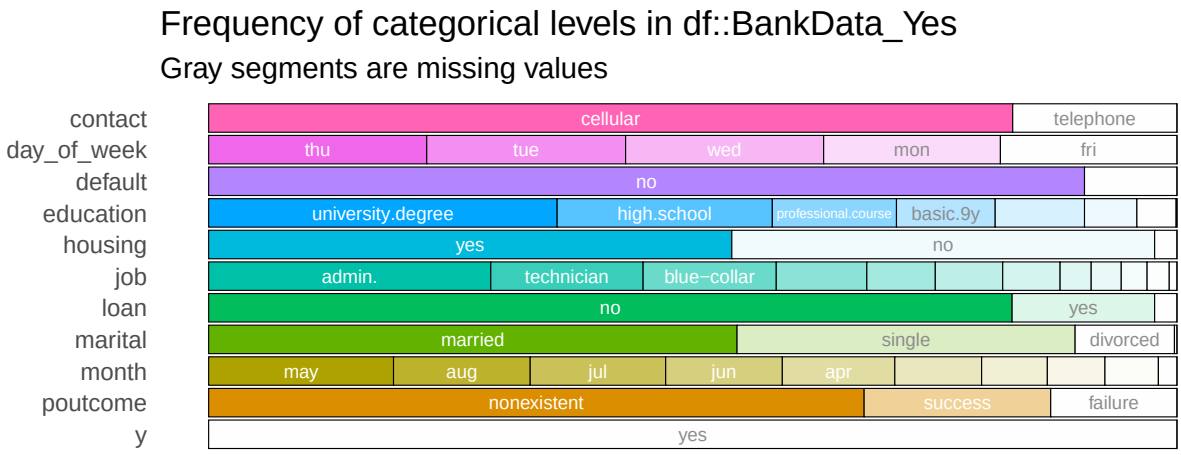

Figure 1.2.1: Frequency of categorical levels with subscribing the term deposit

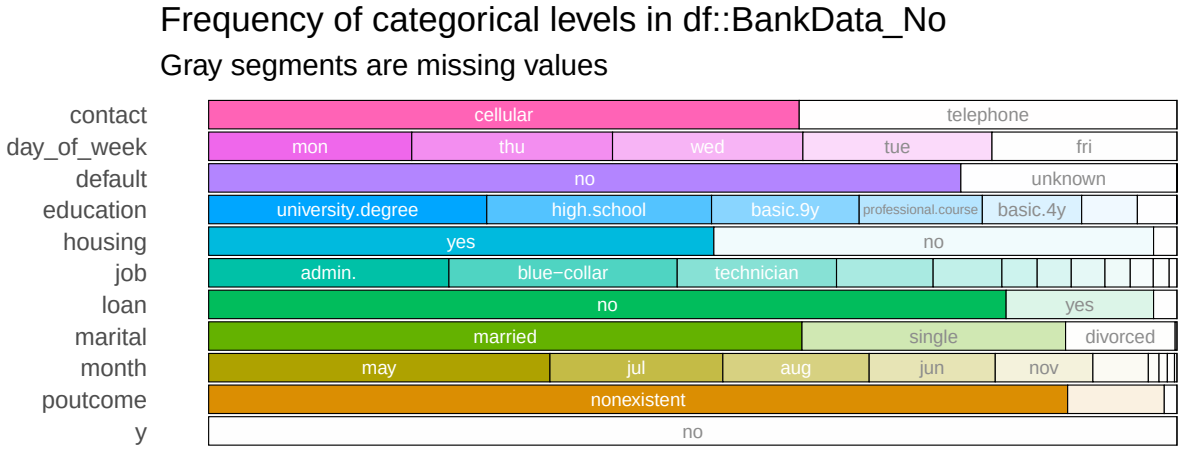

Figure 1.2.2: Frequency of categorical levels without subscribing the term deposit

In this section, we provide a preliminary description of the factor and numeric variables. We are interested in the distribution of related variables, the percentages for each factor variable and the density map for

the numeric variables. Now we can observe the relationship between different levels of factor variables and whether the customers commit to a long term deposit. Figure 1.2.1 and Figure 1.2.2 show that the proportion of people using cellular is larger among those who say "yes" to a long term deposit. This suggests that people using cellular are more inclined to commit to a long term deposit than people using a telephone. With respect to the variable "default", the proportion of people who do not have credit is significantly larger than those who say "no" to the deposit. Therefore, we can see that people without credit are more inclined to say "yes" to a long term deposit. As for the education variable, the proportion of people who have a university degree is significantly higher than that of those who say "no". Ultimately, this suggests that those with a university degree are more likely to say "yes". With respect to the job variable, blue-collar accounts for more people who say "yes" than people who do not. The proportion of government personnel among those who accept the long term deposit is higher than the proportion who do not accept the long term deposit. Besides, the proportion of married people who refused the deposit is significantly higher than the proportion of people who agreed to the deposit. In terms of months, the percentage of people who agreed to the deposit in May was significantly larger than those who did not agree to the deposit. With respect to the poutcome variable, people who have previously accepted the deposit plan are more likely to accept another deposit plan. Examining the housing, loan, and day of week variables, we see that the ratio of these factor levels is roughly the same whether the the customer accepts or does not accept the long term deposit. Therefore, the attitude of a customer saying "yes" or "no" to a long term deposit may have nothing to do with housing, loan, and day of the week. According to the Figure 1.2.3, we can inspect the relationship between the number of variables and whether a customer will say "yes" to a long term deposit. In Figure (A), we see that, regardless of whether the customer agrees to a long term deposit, the distribution of age is roughly the same. We can speculate that the year-old collar has no obvious relationship with the decision a customer makes on the deposit offer. In Figure (B), the last contact duration of those who accept the long term deposit offer is significantly longer than those who say "no" to the deposit offer. We can speculate that the longer the communication time, the more likely that people will say "yes." In Figure (C), the employment variation rate is close to 1 meaning that the density is significantly lower than those who do not want the long term deposit. This implies that the greater the value of employment variation rate, the more likely it is that people will refuse to order. In Figure (D), we see that people are more willing to refuse the long term deposit when the consumption price index is close to 94. Figure (E) indicates when the confidence index is close to -37, -43, -47, people are more willing to refuse the long term deposit. Figure (F) indicates when euribor 3-month rate is close to 5, people are more prone to say "no" to a long term loan.

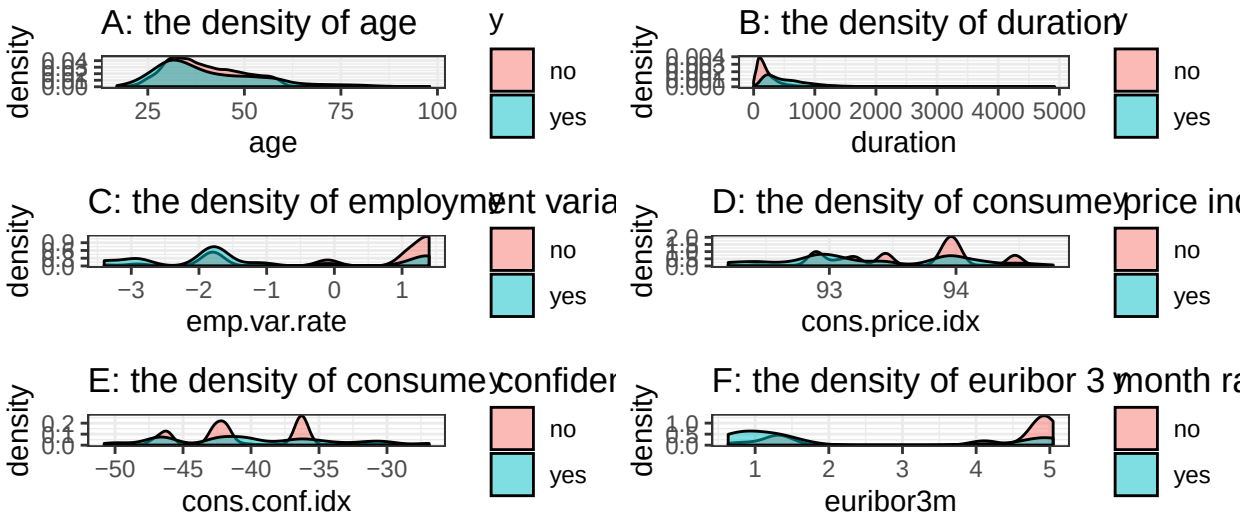

Figure 1.2.3: Density of numerical variables. (A). The density of age; (B). The density of duration; (C). The density of employment variation rate; (D). The density of consume price index; (E). The density of consume confidence index; (F). The density of euribor 3 month rate

2. Statistical Analysis

To build the classification model, the whole data set was split randomly into a training set (70% of the whole dataset) and a testing set (30% of the whole dataset). The three models we propose for the Bank Marketing Project are:

Logistic Regression Model: We develop logistic regression model by 10-fold cross-validation. The training dataset is splitted into 10 subsets. For each validation, 9 out of 10 subsets are used to fit a logistic regression model, and the remaining one subset is used to calculate the accuracy. The final model would be the best one in 10 fitted logistic regression model with highest accuracy. The logistic regression model is $\log \frac{P}{1-P} = \beta_0 + \sum_{i=1}^{p} \beta_i X_i$ where P is the probability that the client says "yes"(make the subscribution), X_i's are the selected variables in the logistic regression model, β_i is the coefficient of X_i and p is the number of selected variables. Assumptions of logistic regression model are (1) the response variable to be binary. (2) the observations to be independent of each other. (3) little or no multicollinearity among the independent variables. (4) linearity of independent variables and log odds. (5) the sample size is large enough.

Random Forest: The random forest model applies the technique of bagging to decision trees. Given a training set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i$ is the documented characteristics vector for the i-th subject in the bank marketing dataset($i = 1, 2, \ldots, 28831$), the algorithm fits 20 independent trees to these samples. The output for each decision tree is the probability of the clients saying "yes" to the long term deposit. Then the predictions for new subject $\mathbf{x}'$ can be made by averaging the predictions from all decision trees $\hat{f}(\mathbf{x}') = \frac{1}{20} \sum_{b=1}^{20} f_b(\mathbf{x}')$. The number of predictors in our model is determined by 10-fold cross-validation. We choose the number of predictors in the final model that maximizes the accuracy. No formal distributional assumption is made for the random forest method.

Gradient boost tree by xgboost: Given the training dataset $\mathbf{X}, \mathbf{Y}$, we develop the gradient boost tree by reconstructing the unknown functional dependence $\mathbf{X} \to \mathbf{Y}$ with some estimated model $\hat{f}(\mathbf{X})$, such that the empirical binomial loss function $L = \sum_{i=1}^{28831} \Psi(\mathbf{y}_i, \hat{f}(\mathbf{x}_i))$ is minimized (where Ψ is the empirical binomial loss function for our project). We achieve this goal by the iterative procedure, which starts with a prespecified decision tree. In each step, the procedure builds a new decision tree based on the previous tree to improve the result by minimizing the empirical binomial loss function as much as possible. The procedure stops when the decrease of loss function is less than a specific amount. In R, the `xgboost` package can efficiently develop the gradient boost tree. The assumption for this method is that the binomial loss function's subgradients are well defined and it is automatically satisfied in our project. [1.][2.] There are two hyperparameters we need to determine by cross validation. They are the max depth of the trees and the colomun sampling ratio by trees.

3. Results

3.1 Logistic Regression

For interpretability, the stepAIC procedure is applied to reduce the number of variables in logsitic regression. The fitted logistic regression model by 10-fold cross-validation is $\log \frac{P}{1-P} = \beta_0 + \sum_{i=1}^{9} \beta_i X_i$ where X_1(age), X_2(marital status), X_3(education), X_4(has housing loan), X_5(last contact duration), X_6(employment variation rate), X_7(consumer price index), X_8(consumer confidence index) and X_9(euribor 3 month rate) are the selected variables. The coefficients of these variables are shown in Table 5.1.1 in Appendix 5.1. In particular, the log ratio will increase 0.009488 when age increases one unit given other variables fixed. The log ratio will increase 0.004574 if the duration increases one unit given other variables fixed. The log ratio will increase 1.123 when the consumer price index increases one unit given other variables fixed. The log ratio will increase 0.006449 when consumer confidence index increases one unit given other variables fixed. The log ratio will decrease 0.641 if the employment variation rate increases one unit given other variables fixed and the log ratio will decrease 0.3547 when euribor3m variables add one unit given others variables fixed.

The confusion matrix is given by Table 3.1.1. The classifier makes a total of 12356 predictions in the test set. Out of these cases, the logistic regression model predicts "yes" 685 times and "no" 11671 times. In reality, 1392 clients actually subscribe the deposit and 10964 clients do not subscribe the client. In particular, there are 436 clients that we predict "yes"(they will subscribe the deposit) and they do subscribe the deposit.

There are 10715 clients that we predict "no"(they will not subscribe the deposit) and they do not subscribe the deposit. There are 249 clients that we predict "yes" but actually they do not subscribe the deposit and there are 956 clients that we predict "no" but actually they actually subscribe the deposit. The sensitivity is 0.3132. It measures how often does the random forest predict a client as "yes" when the client actually subscribes the deposit. The specificity is 0.9772. It measures how often does the logistic model predict a client as "no" when the client actually does not subscribe the deposit. Besides, the precision of this logistic model is 0.9205. It measures when the prediction is "yes", how often is it correct. The AUC which measures the goodness of classification, of our random forest is 0.9163817. It is quite close to 1 and thus, the logistic regression model seems to be good.

	Actual class: Yes	Actual class: No
Prediction: Yes	436	249
Prediction: No	956	10715

Table 3.1.1 Confusion Matrix for Logistic Regression by Cross-Validation

Now we check the model assumptions for logistic regression: (1) In the logistic regression model, the assumptions of binary response and large sample size are automatically satisfied. (2) Since the bank communicated with customers through telephone calls independently, the assumption of independence is also roughly satisfied. (3) The VIFs of 5 numeric variables (age, emp.var.rate, cons.price.idx, cons.conf.idx and euribor3m) in the model are calculated to detect the multicolinearity. They are 1.0085, 1.0114, 2.6824, 2.9288, 1.2967 and 2.4287 respectively. These VIF are all less than 10. This indicates there is no strong multicolinearity among those variables. (4) Pearson correlations are calculated to detect the linearity of independent variables and log odds. The Pearson correlation between log odds and those variables are -0.0036(ages), 0.7884(duration), -0.6632(employment variation rate), -0.5324(consumer price index), -0.0684(consumer confidence index) and -0.6448(euribor 3 month rate). The results shows that the variables age and consumer confidence index are not linear with the log odds. These two variables need to be carefully considered in the future analysis.

3.2 Random Forest

The random forest is one of our alternative approaches in our project. The plot of the accuracy with respect to the number of the predictors is given by Figure 3.3.1 (Top). The number of the predictors in our forest is 10 with the highest average accuracy 91.25%.

	Actual class: Yes	Actual class: No
Prediction: Yes	668	369
Prediction: No	724	10595

Table 3.2.1 Confusion Matrix for Random Forest Data

The confusion matrix is given by Table 3.2.1. The random forest predicts "yes" 1037 times and "no" 11319 times. In particular, there are 668 clients that we predict "yes" and they do subscribe the deposit. There are 10595 clients that we predict "no" and they do not subscribe the deposit. There are 369 clients that we predict "yes" but actually they do not subscribe the deposit and there are 724 clients that we predict "no" but actually they actually subscribe the deposit. The sensitivity is 0.4989. It measures how often does the random forest predict a client as "yes" when the client actually subscribes the deposit. The specificity is 0.9663. It measures how often does the random forest predict a client as "no" when the client actually does not subscribe the deposit. Besides, the precision of this random forest is 0.9376. It measures when the prediction is "yes", how often is it correct. The AUC which measures the goodness of classification, of our random forest is 0.9311. It is quite close to 1 and thus, the random forest model seems to be good.

3.3 XG Boost

The gradient boosting tree by xgboost is our another alternative approach in our project. The plot of the accuracy with respect to the max tree depth for subsample ratio of columns equals to 0.5 and 0.7 is given by Figure 3.3.1 (Bottom). The number of the predictors in our forest is 10 with the highest average accuracy of the cross validation is 91.25%.

	Actual class: Yes	Actual class: No
Prediction: Yes	609	279
Prediction: No	783	10685

Table 3.3.2 Confusion Matrix for Gradient Boosting Tree by XGBoost.

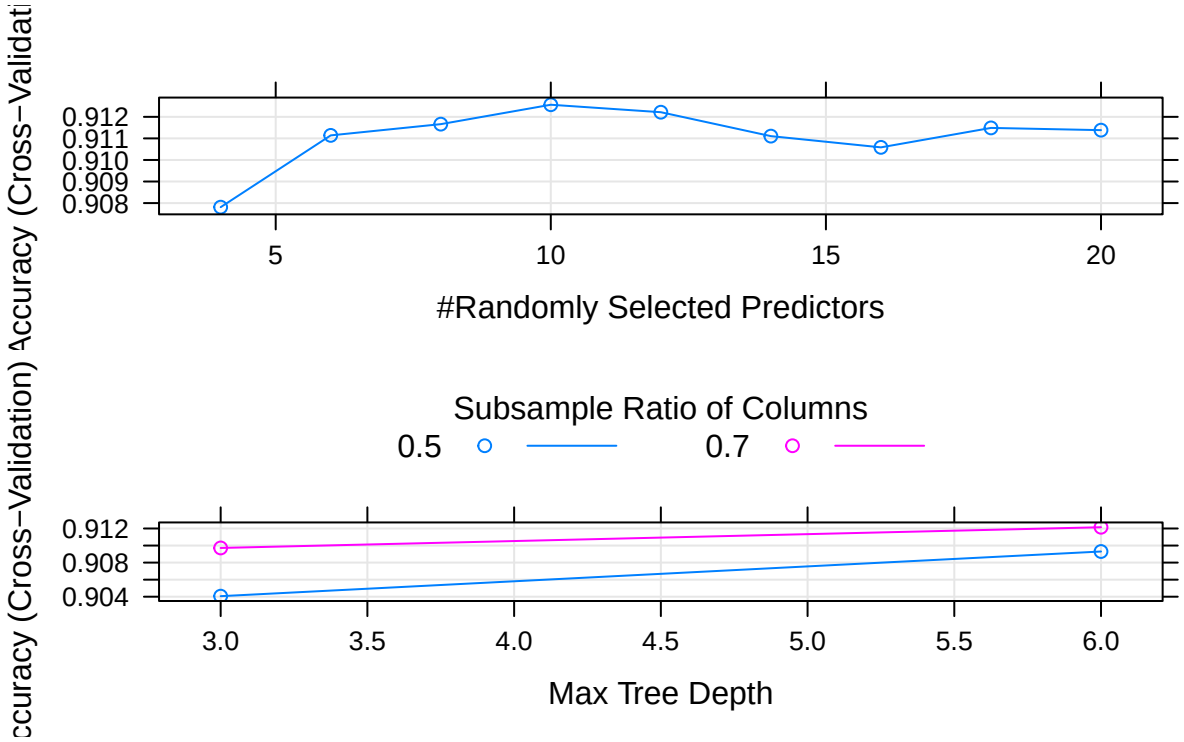

Figure 3.3.1 Top: The plot of the accuracy with respect to the number of the predictors. Bottom: The plot of accuracy of gradient boosting tree model with respect to max tree depth by subsample ratio of columns 0.5 and 0.7.

The confusion matrix is given by Table 3.3.2. The gradient boosting model predicts "yes" 888 times and "no" 11468 times. In particular, there are 609 clients that we predict "yes" and they do subscribe the deposit. There are 10685 clients that we predict "no" and they do not subscribe the deposit. There are 279 clients that we predict "yes" but actually they do not subscribe the deposit and there are 783 clients that we predict "no" but actually they actually subscribe the deposit. The sensitivity is 0.4375 which measures how often does the model predict a client as "yes" when the client actually subscribes the deposit. The specificity is 0.9746. It measures how often does the model predict a client as "no" when the client actually does not subscribe the deposit. Besides, the precision of this model is 0.9269. It measures when the prediction is "yes", how often is it correct. The AUC which measures of the gradient boosting trees is 0.9416. It is quite close to 1 and thus, the gradient boosting tree model seems to be good.

3.4 Model Comparison

We compare above three models (logistic regression, random forest and gradient boosting trees using xgboost) by generating the boxplot of key features of classifiers by these methods. The boxplot shown in Figure 3.5.1 is

based on the resampling results. This methods make it appropriate to compare the different machine learning methods.

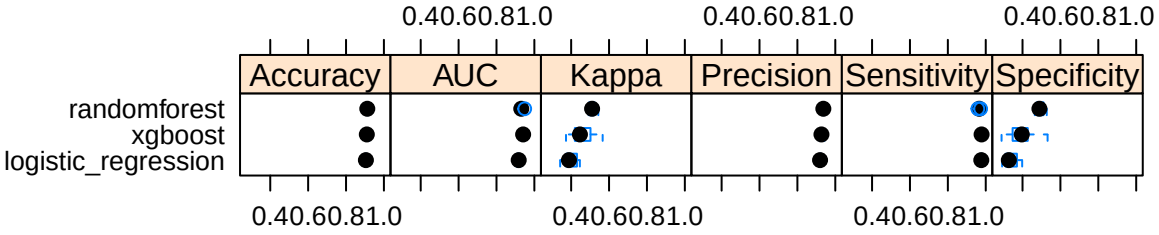

Figure 3.4.1 multi-boxplot of the key features("Accuracy", "AUC", "F1", "Kappa", "Precision", "Sensitivity", "Specificity") by three methods.

Figure 3.4.1 shows that there seems to be no big difference between three models based on Precision, Sensitivity and Accuracy. The specificity and kappa value is largest in the random forest model and smallest in the logistic regression. The specificity of these three models indicates the random forest model has the largest frequency that the random forest predict a client as "no" when the client actually does not subscribe the deposit. Besides, the kappa value of the classifier measures a metric that compares an Observed Accuracy with an Expected Accuracy. It measures how closely the instances classified by the classifier matched the data labeled as ground truth. The medians of kappa statistics for three models are given by $\kappa_1 = 0.3883, \kappa_2 = 0.5105, \kappa_3 = 0.4463$, where κ_1, κ_2 and κ_3 is the median kappa values for logsitic regression, random forest and xgboost method respectively. Based on the kappa statistic, the random forest model is preferred.

4. Conclusion and Discussion

Note that the dataset is highly unbalanced. The proposition of "yes" in response is much lower than that of "no". This issue will reduce the accuracy of our classifier. Therefore, here we are interested in the performance of the random forest model on the under-sampled data. The data are proprocessed by under-sampling the majority class "no" from our original dataset.

	Actual class: Yes	Actual class: No
Prediction: Yes	1276	1586
Prediction: No	116	9378

Table 4.1 Confusion Matrix for Random Forest in undersampled dataset.

The confusion matrix is given by Table 4.1. The undersampled random forest predicts "yes" 2862 times and "no" 9494 times. In particular, there are 1276 clients that we predict "yes" and they do subscribe the deposit. There are 9378 clients that we predict "no" and they do not subscribe the deposit. There are 1586 clients that we predict "yes" but actually they do not subscribe the deposit and there are 116 clients that we predict "no" but actually they actually subscribe the deposit. The accuracy of the random forest model for undersampled data is 0.8623. Thus, random forest on original data and gradient boosting tree models are better than this model in terms of accuracy. This make sense because in undersampling process some information in the full dataset is lost. Besides, due to the computation complexity of random forest on over-sampled dataset, we did not train the corresponding classifier. Oversampling on original dataset might solve the unbalance of the dataset without loosing the information. Besides, the model diagnostic of logistic regression indicates that age and consumer confidence index should be carefully considered when we fit other logistic regression classifiers using these two variables.

5. Appendix

5.1 Detailed Coefficients for Logistic Regression Model

Coefficients:	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-105.9	6.531	-16.218	< 2*e-16	***
age	0.0095	0.0023	4.203	2.63*e-05	***
marital-married	0.0034	0.0783	0.044	0.96524	
marital-single	0.2431	0.0884	2.750	0.00596	**
marital-unknown	0.3506	0.4234	0.828	0.40770	
education-basic.6y	-0.1263	0.1398	-0.903	0.36628	
education-basic.9y	-0.1066	0.1052	-1.014	0.31076	
education-high.school	0.1074	0.0939	1.144	0.25256	
education-illiterate	1.377	0.977	1.408	0.15902	
education-professional.course	0.2731	0.1018	2.684	0.00727	**
education-university.degree	0.4099	0.08925	4.593	4.37*e-06	***
education-unknown	0.2612	0.1275	2.048	0.04051	*
housing-unknown	0.2336	0.1588	-1.471	0.14127	
housing-yes	0.004	0.04687	-0.086	0.93116	
duration	0.00458	0.00009	53.661	< 2*e-16	***
emp.var.rate	-0.641	-0.0769	-8.333	< 2*e-16	***
cons.price.idx	1.123	0.06787	16.543	< 2*e-16	***
cons.conf.idx	0.06449	0.004576	14.122	< 2*e-16	***
euribor3m	-0.3547	0.05836	-6.077	1.22*e-09	***

Table 5.1.1 The Coefficients of Logistic Regression Model

5.2 Session Information

```
print(sessionInfo(), local = FALSE)
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17763)
##
## Matrix products: default
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggpubr_0.2.5        inspectdf_0.0.7     magrittr_1.5
##  [4] ROCR_1.0-7          gplots_3.0.3        MLmetrics_1.1.1
##  [7] xgboost_0.90.0.2    e1071_1.7-3         randomForest_4.6-14
## [10] caret_6.0-85        ggplot2_3.2.1       lattice_0.20-38
## [13] descr_1.1.4         C50_0.1.3
##
## loaded via a namespace (and not attached):
##  [1] tidyr_1.0.2          splines_3.6.2        foreach_1.4.8
##  [4] prodlim_2019.11.13   gtools_3.8.1         Formula_1.2-3
##  [7] assertthat_0.2.1     stats4_3.6.2         yaml_2.2.1
## [10] progress_1.2.2       ipred_0.9-9          pillar_1.4.3
```

```
## [13] glue_1.3.1           pROC_1.16.1          digest_0.6.23
## [16] ggsignif_0.6.0       colorspace_1.4-1     recipes_0.1.9
## [19] cowplot_1.0.0        htmltools_0.4.0      Matrix_1.2-18
## [22] plyr_1.8.5           timeDate_3043.102    pkgconfig_2.0.3
## [25] purrr_0.3.3          xtable_1.8-4         mvtnorm_1.1-0
## [28] scales_1.1.0         gdata_2.18.0         gower_0.2.1
## [31] lava_1.6.6           Cubist_0.2.3         tibble_2.1.3
## [34] farver_2.0.3         generics_0.0.2       withr_2.1.2
## [37] nnet_7.3-12          lazyeval_0.2.2       survival_3.1-8
## [40] crayon_1.3.4         evaluate_0.14        nlme_3.1-142
## [43] MASS_7.3-51.5        class_7.3-15         prettyunits_1.1.1
## [46] tools_3.6.2          data.table_1.12.8    hms_0.5.3
## [49] lifecycle_0.1.0      stringr_1.4.0        munsell_0.5.0
## [52] compiler_3.6.2       inum_1.0-1           caTools_1.18.0
## [55] rlang_0.4.4          grid_3.6.2           iterators_1.0.12
## [58] labeling_0.3         bitops_1.0-6         rmarkdown_2.1
## [61] partykit_1.2-6       gtable_0.3.0         ModelMetrics_1.2.2.1
## [64] codetools_0.2-16     reshape2_1.4.3       R6_2.4.1
## [67] lubridate_1.7.4      knitr_1.28           dplyr_0.8.3
## [70] libcoin_1.0-5        KernSmooth_2.23-16   stringi_1.4.4
## [73] Rcpp_1.0.3           vctrs_0.2.2          ggfittext_0.8.1
## [76] rpart_4.1-15         tidyselect_1.0.0     xfun_0.12
```

5.3 Reference

[1.] Friedman, Jerome H. Stochastic gradient boosting. Computational Statistics and Data Analysis, 38(4):367–378, 2002.

[2.] Liaw, Andy and Wiener, Matthew. Classification and regression by random forest. R News, 2(3): 18-22,2002.

[3.] XGBoost: A scalable Tree Boosting System, Tianqi Chen, Carlos Guestrin, ONR (PECASE) N000141010672, NSF IIS 1258741 and the TerraSwarm Research Center sponsored by MARCO and DARPA.

5.4 Resources

[1.] https://www.r-bloggers.com/dealing-with-unbalanced-data-in-machine-learning/

[2.] https://rpubs.com/fabiorocha5150/decisiontreemodel?fbclid=IwAR23TCDaBPGzCFVGm7Pf44BQkDdwzHhEIUL-oDut8imL1dT3wIvPdXAcOK0

[3.] https://www.hackerearth.com/zh/practice/machine-learning/machine-learning-algorithms/beginners-tutorial-on-xgboost-parameter-tuning-r/tutorial/

[4.] https://www.frontiersin.org/articles/10.3389/fnbot.2013.00021/full

[5.] https://rpubs.com/shienlong/wqd7004_RRookie(Portuguese Bank Marketing Data WQD7004/RRookie/Yong Keh Soon-WQD180065, Vikas Mann-WQD180051, L-ven Lew Teck Wei-WQD180056, Lim Shien Long-WQD180027)

5.4 Github information

https://github.com/BillXu999/Team12_Project4/blob/master/README.md