Prediction of Bank Client's Term-deposit Subscription Behavior Utilizing Supervised Learning Models

Team Members: Hetian Bai (hb1500), Jieyu Wang (jw4937), Zhiming Guo (zg758), Fu Shang (fs1520)

Code Source: https://github.com/gzmkobe/1001_Machine_Learning_Project

1. Introduction

This is a report for *Introduction to Data Science* course team project. In this machine learning project, we are solving a classification problem with the bank marketing data in 2008 from UCI data pool[1]. There are several classification models introduced in class, we apply logistic regression (LR), Support Vector Machine (SVM), decision trees (DT), and random forests (RF) models in this project. The goal is to make automatic predictions of the target variable with the best supervised learning model compared by two performance evaluation metrics: AUC and Lift. This result will contribute to the bank's marketing team by optimizing the process of targeting potential clients with less time and labor costs.

1.1. Business Scenario and Motivation

A Portuguese banking institution attempts to get more of its clients to subscribe to a term deposit. A higher amount of term deposit subscription creates more opportunities for the bank to increase profit, which allows the bank to invest in higher gain financial products and to pay higher interest to its customers. Therefore, we are making binary predictions on target variable that whether a client will subscripe for term deposits with the bank.

1.2. Problem Formulation

- Proactively reach out to bank clients whom with higher likelihood of making subscriptions
- Study the features to gain insights into key drivers to promote clients to make subscriptions
- Create an effective marketing strategy based on data mining and analysis

The predictive model will contribute to the bank's marketing team to identify their potential clients who are more likely to make subscriptions for term deposit. Given a client's information, this model will make a predictive result of whether the customer will subscribe. Then, the marketing team can focus on advertising the bank's term deposit products to clients who are predicted positively. Additionally, the dataset includes some data about when and how often the client was contacted. By data mining, we are able to give suggestions on how to conduct campaigns in a more effective way.

2. Data Exploration

2.1. Predictor Variables

The historical data contains 41,188 instances. Each data instance includes 20 features before data processing, which are categorized into four categories: client's personal information, client–bank relationship, last campaign context, and social and economic indicators.

Table 1. Set of Attributes

Category	**Attributes**	**Description**	**Type**
Client personal information	*age*	Age of the client	Numeric
	job	Client's occupation	Categorical
	marital	Marital status	Categorical
	education	Client's education level	Categorical
Client–bank relationship	*default*	Indicates whether the client has credit in default	Categorical
	housing	Indicates whether the client has a housing loan	Categorical
	loan	Indicates whether the client as a personal loan	Categorical
Last campaign context	*contact*	Type of contact communication	Categorical
	month	Month that last contact was made	Categorical
	day_of_week	Day that last contact was made	Categorical
	duration	Duration of the last contact in seconds	Numeric

	campaign	Number of contacts performed during this campaign for this client (including the last contact)	Numeric
	pdays	Number of days since the client was last contacted in a previous campaign	Numeric
	previous	Number of contacts performed before this campaign for this client	Numeric
	poutcome	Outcome of the previous marketing campaign	Categorical
Social and economic indicators	*empvarrate*	Employment variation rate (quarterly indicator)	Numeric
	conspriceidx	Consumer price index (monthly indicator)	Numeric
	consconfidx	Consumer confidence index (monthly indicator)	Numeric
	euribor3m	Euribor 3-month rate (daily indicator)	Numeric
	nremployed	Number of employees (quarterly indicator)	Numeric

2.2. Target Variable

The target variable *"y"*, is binary and indicates whether the client has subscribed to a term deposit or not. Among the 41,188 customers, 4,640 instances made term deposits and 36,548 customers did not.

3. Data Processing

3.1. Data cleaning

3.1.1. Deal with textual data

a. Convert categorical variables into dummy variables: convert *marital, education, default, housing, loan, contact, poutcome,* and target variable *y* into dummy variables.
b. Convert time data into numeric: convert variables *month (jan/feb/…/dec)* and *day_of_week (mon/tue/…/fri)* into sequence of integers time sequence.

3.1.2. Deal with missing values

a. For SVM and LR: On the one hand, when dealing with missing values, we create a new binary variable named ‘*pdays_mv’* whose value is ‘1’ if there is a missing value in ‘*pdays’*. Instead of using ‘999’ to stand for missing values*,* we take the mean for non-missing values to replace the missing values. On the other hand, we normalized the whole dataset in order to fit the SVM and LR model efficiently.

b. For DT and RF: No additional process on data. Fitted the model directly on the cleaned data set.

4. Feature Exploration

4.1. Correlations

We visualize the correlation matrix (see figure 1) to present the correlations between features themselves as well as the correlation between features and the target variable *“yy”*.

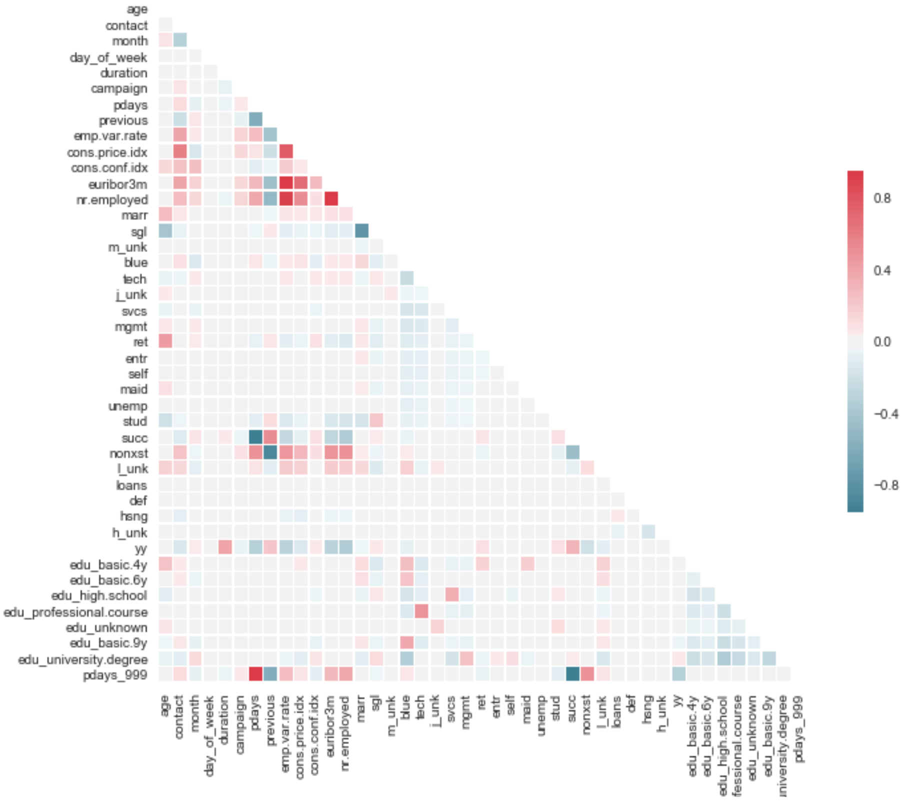

Figure 1. Correlation Matrix for All Features and Target Variable

We find out that there are strong positive correlations between ‘*euribor3m’, ‘consconfidx’, and ‘conspriceidx’*. These economic indices make good sense that these indicators positively represent the performance of the economy in a period of time. In addition, due to the method we deal with missing values in *‘pdays’*, the variable *‘pdays_999’* and *‘pdays’* are strongly correlated. Similarly, ‘succ’ is strongly correlated with ‘pdays’ and ‘pdays_999’.

4.2. Check Nonlinear Relationship

To determine whether the nonlinear relationships exist between variables, we simply fit decision tree model and logistic regression on the same dataset. Decision tree model has AUC_{DT} = 0.8779 and logistic model gives AUC_{LR} = 0.9211. AUC_{LR} is greater than AUC_{DT} indicates there is no significant nonlinear relationship among variables. Therefore, logistic regression model is a good model candidate for this dataset.

4.3. Data Leakage: AUC_{LR} = 0.9211 — "Too good to be true!"

Considering the extremely high AUC from decision tree and logistic regression model, we believe data leakage may exist. The reason is that we aim to predict people's behavior, an AUC over 0.9 is not reasonable. The AUCs we obtained are from models without feature selection and tuning parameters. It is they are too high to be true in this scenario. We plotted the Feature Importance plot (figure 2) to check the importance of each feature, it is clear to see that *'duration'* is much higher than the rest features. Follow this clue, we go back to the original data source then find out this attribute highly affects the output target (e.g., if duration=0 then y=no). Yet, the duration is not known before a call is performed. Also, after the end of the call, target 'y' is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model. The reason we get unreasonably high AUC is stem from data leakage.

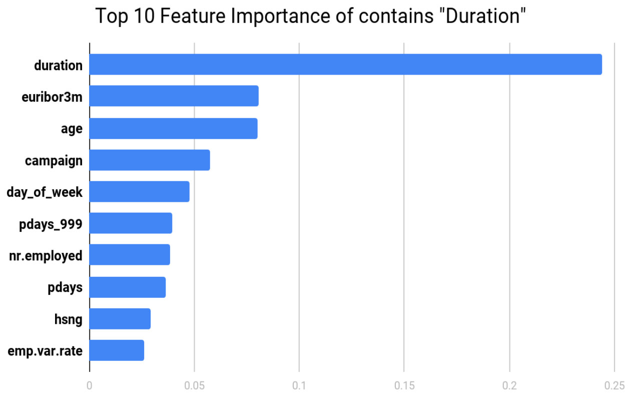

Figure 2. Feature Importance before dropping 'duration'

Based on the above analysis, the leaking feature *'duration'* be removed from the dataset (see figure 3), then we fit the logistic regression model, the $AUC_{LR} = 0.7623$ goes down to a reasonable level, same happens to decision trees $AUC_{DT} = 0.6249$. Still, logistic regression works well as we can see from now.

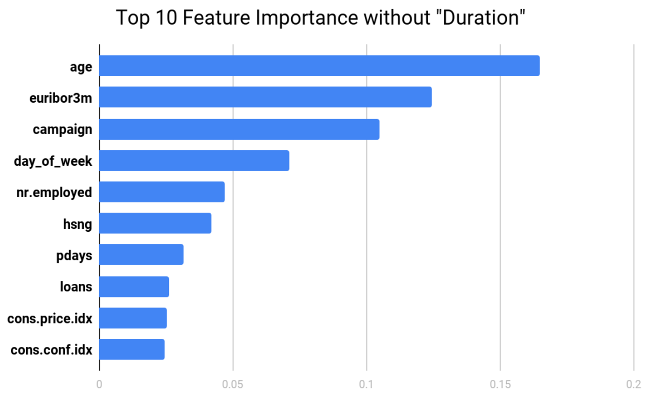

Figure 3. Feature Importance after dropping 'duration'

5. Predictive Modeling and Evaluation

5.1. Modeling Plan

To solve the telemarketing problem, we fit supervised learning models to help with decision making. Specifically, we fit four models in this part and each part includes the discussion of advantages and shortages of each model, as well as the process of tuning parameters to achieve the optimal model performance.

5.2. Modeling Design Parameters

- Algorithms: logistic regression, SVM, decision tree, random forests.
- Hyper-parameters: Each algorithm has its own set of applicable hyper-parameters. In order to find the best parameters with our scoring metric and also to decrease the possibility of overfitting, we take advantage of *GridsearchCV* from *Scikit-learn* in Python to search for the optimal parameters within a range of possible values.

5.3. Evaluation Plan

We split the data into a training set (80%) and a testing set (20%). We do not create a single validation set because we use the method of cross-validation. Also, notice that the data is affected by the order of month. When doing k-fold cross-validation, the fold with most recent data as train set performs always better than those with older data. Thus, we decide to use shuffle K-Fold and set random-state numbers to 42.

5.4. Two Scoring Metrics

1) Area Under Curve (AUC): we need to rank candidates according to the likelihood of clients to make the subscription, and to estimate the probability of a successful outcome for each marketing call. AUC would be a good metric to evaluate model performance by the above criterion.

2) Lift: "Lift" is the most commonly used metric to measure the performance of targeting models in marketing and business applications. Lift measures the performance of a predictive model compared to a baseline without using any models[2]. The formula to calculate lift is listed below:

$$\text{lift} = \frac{(TP/(TP + FN)}{(TP + FP)/(TP + TN + FP + FN)}$$

5.5. Set up Baseline

We assume the marketing team will contact each client in the whole list and the probability of successful subscription rate is 0.1127 (ratio = positive counts / total). In this scenario, AUC equals to 0.5, and Lift is 1.0.

5.6. Modeling

5.6.1. Logistic Regression

1) **Pros and Cons:**

If a dataset has a categorical dependent variable, logistic regression is good to try and runs much faster than other models. To train the LR model, the original dataset has been rescaled and normalized, thus we can use the ‘sag' solver and get a very short training time, whose average is around 0.50 second.

However, LR model is more sensitive to the multicollinearity of the independent variables[3]. If some columns are highly relevant, the weaker ones might get opposite regression signs.

2) Tuning Parameters:

We optimize the performance of LR model by tuning its parameter C, inverse of regularization strength. The baseline model should be LR with default parameter values which is C=1. The mean AUC of baseline model is about 0.77 and Lift score is about 5.7. Figure 4 visualizes the coefficient value for each feature.

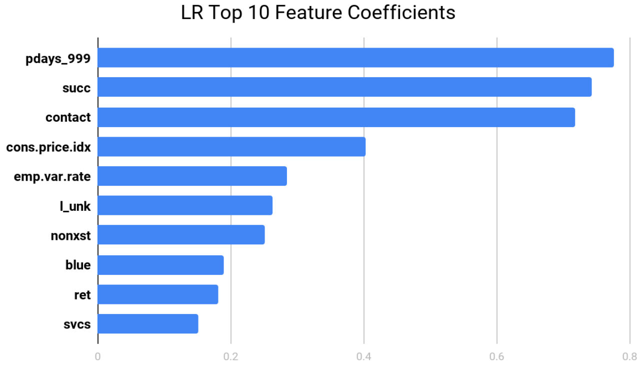

Figure 4. LR Top 10 Feature Coefficients

Train the LR model with a range of C from 10^{-15} to 10^{25}, 5 powers a step, to find the optimal value of C. The result shows an optimal C is possibly near the C=1 point, so we change the range of C into $[10^{-3},10^{9}]$ with 1 power as a step. This time, we get the AUC and Lift Score curves as shown in figure 5 and 6. Based on the curves, we find the best parameters (see table 2):

Table 2. LR Scoring Metrics Performance

Scoring Metric	Best Score	Best Parameters
AUC	0.773	C=1
Lift	6.060	C=100

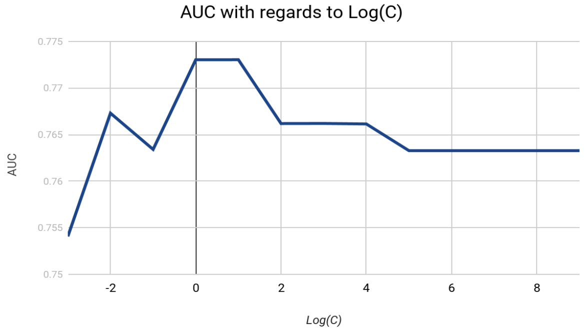

Figure 5. AUC vs. Log(C)

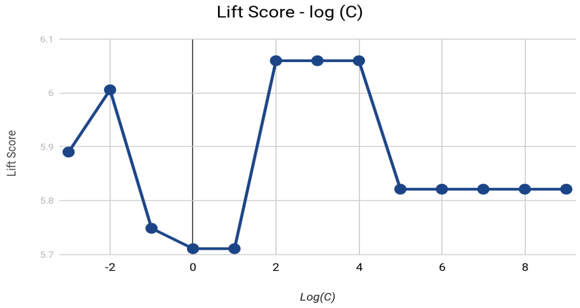

Figure 6. Lift vs. Log(C)

5.6.2. SVM

1) Pros and Cons

Although SVM can be applied to problems with a non-linear hyperplane, the time-complexity of training SVM is not desirable. As discussed in Section 4.2 of *"Support vector machine solvers"*[4], the time complexity depends on R^3 and n*S, where R is the number of free support vectors, n is the number of training samples and S is the number of support vectors. In our case, there are 41,188 instances, which can cause the training time of SVM to be extremely long. Another drawback for SVM is from *scikit-learn*, SVM does not support incremental training, which makes it difficult to refit new models with new incoming data.

2) Tuning Parameters

In practice, SVM with linear kernel performs almost like logistic regression. Also, our dataset has a non-linear hyperplane due to categorical variables. Therefore, we set the kernel of SVM to be "rbf" (radial basis function). The RBF kernel on two samples x and x', is defined as:

$$K(\mathbf{x}, \mathbf{x'}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x'}\|^2}{2\sigma^2}\right)$$

Gamma is a parameter specifically in 'rbf' defining the influence of a single training example reaches. Parameter C controls regularization and thus can be used to avoid overfitting. After searching, we find that the best parameters are in table 3:

Table 3. SVM Scoring Metrics Performance

Scoring Metric	Best Score	Best Parameters
AUC	0.707	C=0.1, gamma=0.1
Lift	6.266	C=0.1, gamma=0.1

5.6.3. Decision Trees

1) Pros and Cons

Decision tree often performs well on imbalanced datasets because their hierarchical structure allows them to learn signals from both classes[5]. The biggest advantage is that it is easy to be interpreted: it makes explicit all possible alternatives and traces each alternative to its conclusion in a single view, allowing for easy comparison among the various alternatives. It can perform feature selection and won't be affected by non-linear relationships between parameters. However, it will create over-complex tree structure causing overfitting. Also, it is unstable since small variations in the dataset yield completely different trees.

2) Tuning Parameters

The base model of decision tree using default parameters (criterion is entropy) has AUC of 0.625. To improve AUC, we use *gridsearch* function to find three parameters (minimal size for the split, minimal leaf size, maximal depth) in decision tree. The optimal combination of these three parameters results in minimum overfitting and highest AUC and Lift.

Inserting the optimal parameters into the decision tree model results feature importance indicating that variables *'nr.employed', 'euribor3m', 'emp.var.rate', 'pdays', 'cons.conf.idx'*, and so on are important features. Refitting the model by adding one feature at a time from the most important to least to see the change of AUC. The best AUC below is obtained by including top 10 important features, which are *'nr.employed', 'euribor3m', 'emp.var.rate',*

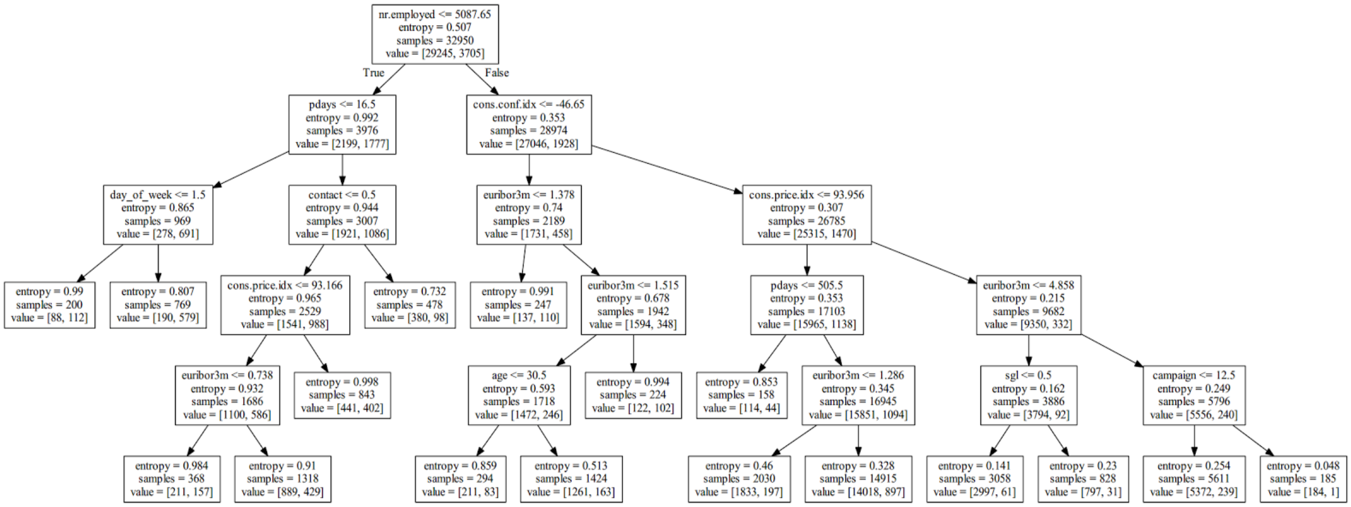
nr.employed <= 5087.65 entropy = 0.507 samples = 32950 value = [29245, 3705]
True
False
pdays <= 16.5 entropy = 0.992 samples = 3976 value = [2199, 1777]
cons.conf.idx <= -46.65 entropy = 0.353 samples = 28974 value = [27046, 1928]
day_of_week <= 1.5 entropy = 0.865 samples = 969 value = [278, 691]
contact <= 0.5 entropy = 0.944 samples = 3007 value = [1921, 1086]
euribor3m <= 1.378 entropy = 0.74 samples = 2189 value = [1731, 458]
cons.price.idx <= 93.956 entropy = 0.307 samples = 26785 value = [25315, 1470]
entropy = 0.99 samples = 200 value = [88, 112]
entropy = 0.807 samples = 769 value = [190, 579]
cons.price.idx <= 93.166 entropy = 0.965 samples = 2529 value = [1541, 988]
entropy = 0.732 samples = 478 value = [380, 98]
entropy = 0.991 samples = 247 value = [137, 110]
euribor3m <= 1.515 entropy = 0.678 samples = 1942 value = [1594, 348]
pdays <= 505.5 entropy = 0.353 samples = 17103 value = [15965, 1138]
euribor3m <= 4.858 entropy = 0.215 samples = 9682 value = [9350, 332]
euribor3m <= 0.738 entropy = 0.932 samples = 1686 value = [1100, 586]
entropy = 0.998 samples = 843 value = [441, 402]
age <= 30.5 entropy = 0.593 samples = 1718 value = [1472, 246]
entropy = 0.994 samples = 224 value = [122, 102]
entropy = 0.853 samples = 158 value = [114, 44]
euribor3m <= 1.286 entropy = 0.345 samples = 16945 value = [15851, 1094]
sgl <= 0.5 entropy = 0.162 samples = 3886 value = [3794, 92]
campaign <= 12.5 entropy = 0.249 samples = 5796 value = [5556, 240]
entropy = 0.984 samples = 368 value = [211, 157]
entropy = 0.91 samples = 1318 value = [889, 429]
entropy = 0.859 samples = 294 value = [211, 83]
entropy = 0.513 samples = 1424 value = [1261, 163]
entropy = 0.46 samples = 2030 value = [1833, 197]
entropy = 0.328 samples = 14915 value = [14018, 897]
entropy = 0.141 samples = 3058 value = [2997, 61]
entropy = 0.23 samples = 828 value = [797, 31]
entropy = 0.254 samples = 5611 value = [5372, 239]
entropy = 0.048 samples = 185 value = [184, 1]

Figure 7. tree structure by optimal parameters *'pdays', 'cons.conf.idx', 'cons.price.idx', 'succ', 'month', 'age', and 'contact'*. The optimal AUC is 0.796 where max_depth is 10, min_samples_leaf is 140, and min_samples_split is 1200. The refitting procedure stops at 10 features because the rest features are not relatively important and barely improve AUC after included. Also, more features included would make the model complex.

Using same tuning procedure, the optimal lift is obtained by the parameters where max_depth is 5, min_samples_leaf is 100, and min_samples_split is 900 in the figure below and the lift is 6.255 by DT (9 features included, which are *'nr.employed', 'cons.conf.idx', 'pdays', 'euribor3m', 'cons.price.idx', 'contact', 'age', ‘day_of_week’, and 'campaign’*). Figure 7 shows the tree structure corresponding to the model above.

5.6.4. Random Forests

1) Pros and Cons

In modern applied machine learning, random forests almost always outperform singular decision trees. Compared to the single decision tree, the random forest can develop various trees with k features selected out of p features from the original dataset. Therefore, it doesn't require feature selection and has a more convincing result in feature importance. It can also reduce the variance for an unbalanced dataset. Nevertheless, random forests have been observed to overfit for some datasets with noisy classification/regression tasks. Unlike single decision trees, random forest models are hard to interpret since the result is a combination of various trees.

2) Tuning Parameters

Besides three parameters from decision tree, there are two additional parameters for random forests need to be tuned, which are "number of trees" and "number of predictors sampled". There are a total of 40 predictors in the dataset, typically, the square root of 40, which is 7 predictors in the case of classification make a good choice.

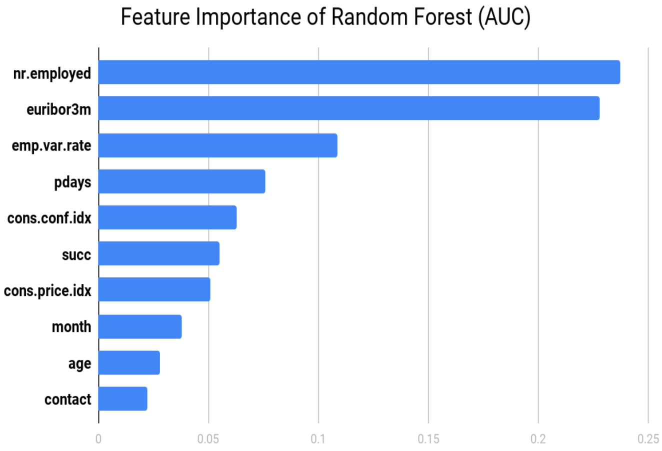

Figure 8. Relative Feature Importance of RF based on AUC

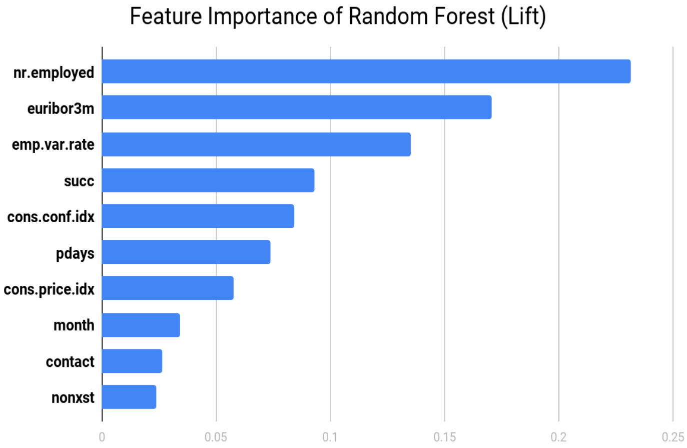

Figure 9. Relative Feature Importance of RF based on Lift

The baseline AUC for the random forest by using default settings in Python is 0.767. By choosing the optimal combination parameters from the dictionary below and setting max_feature ranged from 5 to 10, which means the number of features to consider when looking for the best split is from 5 to 10 (default was 7 due to 40 features). AUC is highest when max_feature is 10. The relative feature importance graph suggests the top 10 important features are *'nr.employed', 'euribor3m', 'emp.var.rate', 'pdays', 'cons.conf.idx', 'succ', 'cons.price.idx', 'month', 'age', and 'contact'*. The optimal AUC is 0.803 where max_depth is 10, min_samples_leaf is 60, min_samples_split is 2, n_estimator is 500. Feature importance plot based on AUC is in figure 8.

Using same procedure, with max_features = 5, the model with the parameters where max_depth is 5, min_samples_leaf is 150, min_samples_split is 2, n_estimator is 500, outputs the highest lift, which is 6.400. The relative important features are *'nr.employed', 'euribor3m', 'emp.var.rate', 'succ', and 'cons.conf.idx'*.

5.7. Model Evaluation

We use Area under Curve (AUC) and lift to evaluate our predictive models. AUC is base rate non-invariant and can be used to compare models from the perspective of the building model. While lift measures how many more positive outcomes we might expect relative to the baseline strategy and is commonly used in marketing scenario. By reading table 4, all model's AUC and lift scores are higher than baseline's. Furthermore, random forest is the best model since both its AUC and lift are the highest among all models, even though it is difficult to explain.. Specifically to solving the marketing problem, lift is an appropriate metric to evaluate models. Lift score equals to 6.400 indicating that using finalized random forest model, we can have 6.4 times enhanced response compared to the population as a whole.

Table 4. Summary Table of Model Performance on AUC and Lift

Metric Model	Baseline	LR	SVM	DT	RF
AUC	0.500	0.773	0.707	0.796	0.803
LIFT	1.000	6.060	6.266	6.255	6.400

6. Business Deployment

6.1. Discussion

- The LR model can provide us with an insight of the impact of each feature on predicting subscription. In figure 4, features in red have positive effects, whereas features in blue have negative effects. The increase of black will decrease the probability of subscription. The length of a bar represents the feature's contribution to explaining target variable.

- From RF's feature importance figure, we can observe that the Social and Economic Indicators, including *emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr.employed,* plays an important role in RF's classification model.

- With the help of RF predictive model, the bank marketing team proactively reach out to clients whom with a higher likelihood of making term deposit subscriptions. By applying the random forests model, marketing team may give calls to the clients labeled ‘1', who are identified as potential customers by the supervised model. The predictive classifier contributes to the bank to get the most subscriptions with fewer costs in terms of time and labor.

6.2. Deployment Issues

When collecting data, all future instances to be predicted should have the exact same features as the original dataset does. Also, the method of preprocessing data should be identical with the code we used to process data on the training set onto the raw data.

6.3. Ethical Considerations

There are two ethical related potential risks in our business deployment. First, we need to store customer’s information safe. If customers’ personal data is leaked, customers’ previate exposed and the bank’s business will also be affected. Second, we need to make sure the screening customers process does not involve any kind of discrimination toward minorities. Although our dataset does not contain customers' race and gender information, it may be the case that customers from a certain race and gender get more positive prediction than others. We need to look closely at the result of our prediction and make sure our model does not lead to discrimination.

6.4. Risk Identified

All models are trained based on the data from Portuguese banks in 2008 when financial crisis dramatically changed the business of banking market. If similar financial crisis occurs in future, this model could be applied for Portuguese banks during hard-hitor recession. Otherwise, we need to develop new models based on the data in peaceful period or slow-recovery period. Portuguese bank during economic-stability period and other banks will take risks using this model.

Meanwhile, the important features are resulted from random forests based on the best parameters. This does not imply that other features are not important. Different models with different evaluation metrics would conclude to completely different important features. Therefore, risk would be higher if we consider important features by trees model only.

6.5. Pipeline

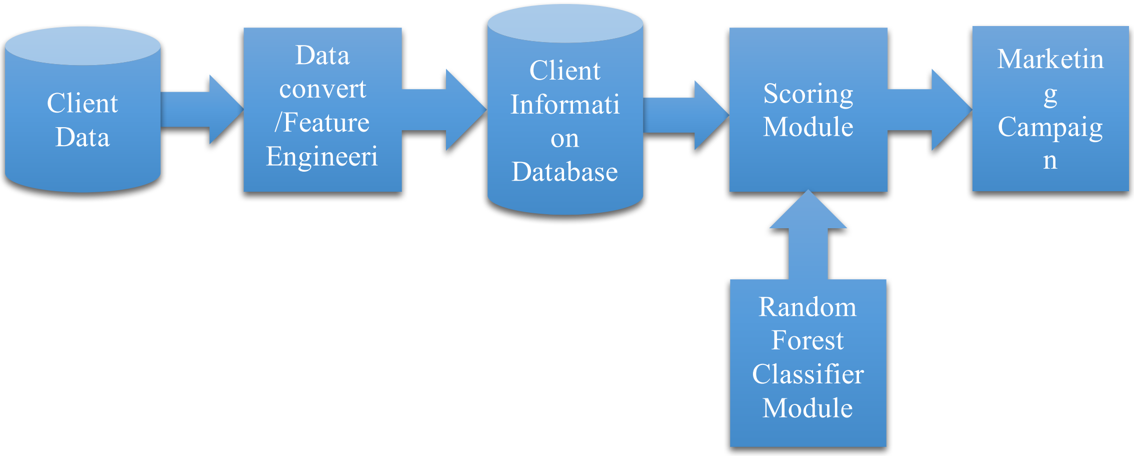
Client Data
Data convert /Feature Engineeri
Client Informati on Database
Scoring Module
Marketin g Campaig n
Random Forest Classifier Module

Figure 10. A simple production scoring architecture

A Brief Description of Components in the Pipeline system in Figure 10:

- Client Data - These are the raw data that will ultimately be processed into our data
- Data Convert/ Feature Engineering Pipeline – In this pipeline, using the identical code we used to process data on the training set onto the raw data.
- Client Information Database: Processed data stored in Database for future analysis
- Random Forest Classifier Module - This is the stored optimal parameters that we obtained from model validation. Apply this parameter data into Predictive Model module in Scoring with Predictive Model Pipeline
- Scoring Module - This module takes in structured features of a user and a Random Forest model to label each instance (label ‘1’ or ‘0’) and obtain the probability.
- Marketing Campaign – This is the prediction result from our analysis, based on that marketing team will contact those clients labeled as ‘1'

Appendix

[1] [Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

[2].Coppock, D. S. (2002). Why Lift? Data Modeling and Mining. Information Management Online.

[3] Aguilera, A. M., Escabias, M., & Valderrama, M. J. (2006). Using principal components for estimating logistic regression with high-dimensional multicollinear data. Computational Statistics & Data Analysis, 50(8), 1905-1924. doi:10.1016/j.csda.2005.03.011

[4] Platt, J. C. (2009). Advances in neural information processing systems 20: proceedings of the 2007 conference. La Jolla, CA: Neural Information Processing Systems Foundation.

[5] How to Handle Imbalanced Classes in Machine Learning. (2017, September 16). Retrieved December 04, 2017, from https://elitedatascience.com/imbalanced-classes

Work Contibution

Name	Hetian Bai	Fu Shang	Jieyu Wang	Zhiming Guo
Contribution	Data Preprocessing	LR	SVM	Tree-based model