Applying Machine Learning to the Development of Prediction Models for Bank Deposit Subscription

Sipu Hou, California State University, East Bay, USA
Zongzhen Cai, California State University, East Bay, USA
Jiming Wu, California State University, East Bay, USA
Hongwei Du, California State University, East Bay, USA
Peng Xie, California State University, East Bay, USA

ABSTRACT

It is not easy for banks to sell their term-deposit products to new clients because many factors will affect customers’ purchasing decisions and because banks may have difficulties in identifying their target customers. To address this issue, the authors use different supervised machine learning algorithms to predict if a customer will subscribe a bank term deposit and then compare the performance of these prediction models. Specifically, the current paper employs these five algorithms: naïve bayes, decision tree, random forest, support vector machine, and neural network. This paper thus contributes to the artificial intelligence and big data field with an important evidence of the best performed model for predicting bank term deposit subscription.

KEYWORDS

Decision Tree, Machine Learning, Naïve Bayes, Neural Network, Random Forest, Support Vector Machine

INTRODUCTION

In the past few decades, banks have experienced some problems of selling their term-deposit products to new clients (Elsalamony, 2014). This is mainly due to the fact that it is difficult for banks to figure out who are their target customers and that many factors will affect a customer’s decision to purchase a term deposit (Moro et al., 2014). Meanwhile, more and more customers start to complain about the irrelevant phone calls received from banks. To address these business issues, banks start to leverage their huge customer data to gain insight into customer behavior and buying preferences, and thus to improve their marketing effectiveness (Wikipedia, 2018a). This strategic business initiative relies on artificial intelligence and Big Data technologies.

Artificial intelligence refers to using computer systems to perform tasks that normally require human intelligence and can be considered as a concept broader than machine learning, even though they are often used interchangeably (R, n.d.). Nowadays, more and more organizations are employing artificial intelligence and Big Data to gain an edge over the rest of the market. There are a variety of machine learning based predictive techniques to be used to predict if customers will subscribe a term deposit. Many researchers consider machine learning as one of the most useful research methodologies...
to transform data into intelligent action and to capture the meaningful information and hidden patterns from the historic datasets (Wikipedia, 2018b).

In this paper, we use machine learning with R to develop prediction models to forecast whether customers will purchase bank term-deposit products. Five different machine learning algorithms have been employed: Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and Neural Network (NN). A real dataset from Portuguese bank has been used for this research. The dataset includes 45221 observations and 20 variables. Below, we present the details about the design and implementation of this research.

RESEARCH METHODOLOGY

A time deposit or term deposit is a deposit with a specified period of maturity and earns interest (Towards Data Science, n.d.). It is a money deposit at banking institution that cannot be withdrawn for a specific term or period (unless a penalty is paid) (Towards Data Science, n.d.). Banks raise the capitals through engaging in direct marketing campaigns to sell their deposits products to clients. Machine learning based predictive techniques can help banks determine which customers are more likely to subscribe a term deposit. Therefore, it is worthwhile for researchers to analyze historic datasets by making a prediction model to forecast whether clients will subscribe a term deposit (Elsalamony, 2014).

From the World Wide Web, we found a dataset about direct marketing campaigns of Portuguese banks. The dataset can be downloaded from the UCI Machine Learning Repository at https://archive.ics.uci.edu/ml/datasets/bank+marketing. This direct marketing campaign dataset was collected from 2008 to 2013 by making phone calls to contact clients to collect their information and by calling the same client again to assess if the client will subscribe a term deposit or not (Lantz, 2015).

DATA DESCRIPTION

This bank marketing dataset includes 45211 observations, each with 20 attributes. The target y attribute shows the result if clients will subscribe the term deposit or not. The target feature is coded as “yes” to indicate client will subscribe a term deposit while “no” means client will not subscribe the term deposit. The 20 attributes are described below:

1. **Age**: The age of the customer (numeric);
3. **Marital**: Marital status (categorical: “divorced”, “married”, “single”, “unknown”; note: “divorced” means divorced or widowed);
4. **Education**: (categorical: “basic.4y”, “basic.6y”, “basic.9y”, “high.school”, “illiterate”, “professional-course”, “university.degree”, “unknown”);
5. **Default**: Has credit in default? (categorical: “no”, “yes”, “unknown”);
6. **Housing**: Has housing loan? (categorical: “no”, “yes”, “unknown”);
7. **Loan**: Has personal loan? (categorical: “no”, “yes”, “unknown”);
8. **Contact**: Contact communication type (categorical: “cellular”, “telephone”);
9. **Month**: Last contact month of year (categorical: “jan”, “feb”, “mar”, ..., “nov”, “dec”);
10. **Day_of_week**: Last contact day of the week (categorical: “mon”, “tue”, “wed”, “thu”, “fri”);
11. **Duration**: Last contact duration, in seconds (numeric). Important note: 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 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;

12. **Campaign**: Number of contacts performed during this campaign and for this client (numeric, includes last contact);

13. **pdays**: Number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted);

14. **Previous**: Number of contacts performed before this campaign and for this client (numeric);

15. **poutcome**: Outcome of the previous marketing campaign (categorical: “failure”, “nonexistent”, “success”);

16. **emp.var.rate**: Employment variation rate - quarterly indicator (numeric);

17. **cons.price.idx**: Consumer price index - monthly indicator (numeric),”success”;

18. **cons.conf.idx**: Consumer confidence index - monthly indicator (numeric) # social and economic context attributes;

19. **euribor3m**: euribor 3 months rate - daily indicator (numeric);

20. **nr.employed**: Number of employees - quarterly indicator (numeric);

21. **Output variable (desired target)**: Has the client subscribed a term deposit? (binary: “yes”, “no”).

The step in the machine learning process after we collected the data involves examining data in detail. The better we understand our dataset, the more proper machine learning algorithm we can match to solve the classification problems. Below we describe the dataset in details.

**Data Preparation**

Before jumping to the method of manipulating missing, we first need to know the reason why data goes missing. This is important because the missing data may bias our results. After we have a good understanding of the missing data, we can then choose an appropriate approach to deal with it. Normally, there are three mechanisms causing data missing. The first is missing at random (MAR), which means that the propensity for a data point to be missing is not related to the missing data, but related to some of the observed data. This means that most missingness is not completely at random. The second is missing completely at random (MCAR), which refers to the fact that a certain value is missing has nothing to do with its hypothetical value and with the values of other variables.

The last is known as missing not at random (MNAR) (Kodali, 2016). Two possible reasons for MNAR are that the missing value depends on the hypothetical value (e.g. People with high salaries generally do not want to reveal their incomes in surveys) or missing value is dependent on some other variable’s value (e.g. Let’s assume that females generally don’t want to reveal their ages and thus the missing value in age variable is impacted by gender variable).

For the first two mechanisms, it is safe to remove the data with missing values depending upon their occurrences, whereas for the third mechanism, removing observations with missing values can produce a bias in the model.
For the current bank deposit dataset, we found that the “unknown” status data exists in several attributes, which are “marital” attribute, the “default” attribute, the “housing” attribute and the “loan” attribute, etc. They all belong to categorical data type. For the “default” attribute, the total amount of “yes” response is only three clients. However, the number “unknown” status is quite large, which is 8,598 in total. In this case, we can’t make imputation because of the extremely small population of “yes” response. Therefore, we decided to keep the “unknown” status as a new category and use it in my algorithm. For the other observations with missing values, it is safe to exclude them.

After the removal, now the dataset have 38245 observations, which is split into training data and testing data. We use stratified sampling to split the data, so that distribution of the outcome within training and testing datasets is preserved. We split the data with the ratio of 66% (25243) for training the model and 34% (or 13002) for testing the model.

**BUILDING THE MODELS**

1. **Naïve Bayes**

The Naïve Bayesian classifier is based on Bayes’ theorem with the independence assumptions between predictors. A Naïve Bayesian model is easy to build, with no complicated iterative parameter estimation making it particularly useful for very large datasets (Saedsayad, n.d.). Using the class label from the training data, Bayes method learns the conditional attribute and then computes the probability of a class value, given the particular instance. Despite its simplicity, the Naïve Bayesian classifier often
predicts the class value with the highest probability and is widely used because it often outperforms more sophisticated classification methods (Saedsayad, n.d.).

We start with loading the necessary library and build an initial model with all features and then evaluate the model on the testing data. The results are shown in Table 4.

<table>
<thead>
<tr>
<th>age</th>
<th>job</th>
<th>marital</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. : 17.00</td>
<td>admin.</td>
<td>divorced: 4612</td>
</tr>
<tr>
<td>1st Qu. : 32.00</td>
<td>blue-collar.</td>
<td>married : 24028</td>
</tr>
<tr>
<td>Median : 38.00</td>
<td>technician.</td>
<td>single : 11568</td>
</tr>
<tr>
<td>Mean : 40.02</td>
<td>services.</td>
<td>unknown : 80</td>
</tr>
<tr>
<td>3rd Qu. : 47.00</td>
<td>management.</td>
<td></td>
</tr>
<tr>
<td>Max. : 98.00</td>
<td>(Other)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>education</td>
<td>default</td>
<td>housing</td>
</tr>
<tr>
<td>Min. : 12168</td>
<td>no</td>
<td>loan</td>
</tr>
<tr>
<td>high.school</td>
<td>9515</td>
<td></td>
</tr>
<tr>
<td>basic.9y</td>
<td>6045</td>
<td></td>
</tr>
<tr>
<td>basic.4y</td>
<td>4176</td>
<td></td>
</tr>
<tr>
<td>basic.6y</td>
<td>2292</td>
<td></td>
</tr>
<tr>
<td>(Other)</td>
<td>1749</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>contact</td>
<td>month</td>
<td>day_of_week</td>
</tr>
<tr>
<td>Min. : 26144</td>
<td>may</td>
<td>duration</td>
</tr>
<tr>
<td>1st Qu. : 32588</td>
<td>jul.</td>
<td></td>
</tr>
<tr>
<td>Median : 6178</td>
<td>aug.</td>
<td></td>
</tr>
<tr>
<td>Mean : 5318</td>
<td>jun.</td>
<td></td>
</tr>
<tr>
<td>3rd Qu. : 8090</td>
<td>nov.</td>
<td></td>
</tr>
<tr>
<td>Max. : 4101</td>
<td>Apr.</td>
<td></td>
</tr>
<tr>
<td>(Other)</td>
<td>2632</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>campaign</td>
<td>previous</td>
<td>outcome</td>
</tr>
<tr>
<td>Min. : 1000</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>1st Qu. : 1000</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>Median : 2000</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>Mean : 2568</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>3rd Qu. : 3000</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>Max. : 5600</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>emp.var.rate</td>
<td>cons.price.idx</td>
<td>cons.conf.idx</td>
</tr>
<tr>
<td>Min. : 3.40000</td>
<td>92.20</td>
<td>euribor3m</td>
</tr>
<tr>
<td>1st Qu. : 3.80000</td>
<td>93.08</td>
<td></td>
</tr>
<tr>
<td>Median : 3.10000</td>
<td>93.75</td>
<td></td>
</tr>
<tr>
<td>Mean : 0.08189</td>
<td>93.58</td>
<td></td>
</tr>
<tr>
<td>3rd Qu. : 3.40000</td>
<td>93.99</td>
<td></td>
</tr>
<tr>
<td>Max. : 3.40000</td>
<td>94.77</td>
<td></td>
</tr>
</tbody>
</table>

The initial model yields quite decent results, especially for the prediction for ‘no’. We see that the misclassified prediction rate for ‘no’ is 5.21% \((570/(570+10375))\), which is quite small. However, the misclassified prediction rate for ‘yes’ is 57.37% \((1180/(1180+877))\). The overall accuracy rate is 86.54% and the sensitivity rate is 89.79% \((10375/(10375+1180))\). These two rates are both excellent.

(2) Decision Tree

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables (Analytics Vidhya, 2016). A decision tree is just a flowchart with several nodes and conditional edges. Each non-leaf node represents a conditional test on one of the features and each edge represents an outcome of the test. Each leaf node represents a class label where predictions are made for the final outcome. Paths from the root to all the leaf nodes give us all the classification rules. Decision trees are easy to represent, construct, and understand. However, the drawback is that they are very prone to overfitting and often these models do not generalize well. We will follow a similar analytics pipeline as before, to build some models based on decision trees (R, n.d.).

We start with loading the necessary dependencies and test data features. We then build a decision tree model with all the features using training data and finally, we predict and evaluate the model on the test data. The results are also excellent. As shown in the following decision tree output, the overall accuracy rate is 91.82% and sensitivity rate is 96.55%. The misclassified prediction rate for
‘no’ is 5.94% \( \frac{664}{11156+664} \), which is quite small. However, the misclassified prediction rate for ‘yes’ is 33.76% \( \frac{399}{399+783} \).

To greatly improve our model’s predictive ability, we can produce numerous trees and combine the results. The logic of the algorithm can be described briefly as follows. At any point in time, each tree in the ensemble of decision trees is built from a bootstrap sample, which is basically sampling with replacement. This sampling is done on the training dataset. During the construction of the decision tree, the split that was earlier being chosen as the best split among all the features is not valid anymore. Now the best split is always chosen from a random subset of the features each time. This introduction of randomness into the model increases the bias of the model slightly but decreases the variance of the model greatly which prevents the overfitting of models, which is a serious concern in the case of decision trees. Overall, this process will yield much better general decision tree models (R, n.d.).

(3) Random Forest

We start our analytics pipeline by loading the necessary dependencies into the R-studio system and then build the random forest training model with all the features. Using the importance() function, we can view the importance of each variable in the random forest algorithm as follows. This gives us the following plot showing the importance of different features.

Next, we perform the predictions on the test data and obtain the results as follows. The results are also very good, showing that the overall accuracy rate is 91.89% and the sensitivity rate is 96.58%.

(4) Support Vector Machine (SVM)

SVM has several different kernels that can be used to achieve machine learning based data analytics. These include regular linear kernel, polynomial kernel, radial basis function (RBF), and several others. They are based on the following algorithms:

Linear kernel: \( K(x_i, x_j) = x_i \cdot x_j \),
Polynomial kernel: \[ K(x_i, x_j) = \left( x_i \cdot x_j + 1 \right)^d \]

Sigmoid kernel: \[ K(x_i, x_j) = \tanh\left( \kappa x_i \cdot x_j - \delta \right) \]

Gaussian RBF kernel: \[ K(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} \]

The main principle behind these non-linear kernel functions is that even if linear separation is not possible in the original feature space, they enable the separation to happen in a higher dimensional transformed feature space where we can use a hyperplane to separate the classes. An important thing to remember is the curse of dimensionality that applies here. Since we may end up working with higher dimensional feature spaces, the model generalization error increases and the predictive power of the model decreases. But if the dataset is large enough, it will still perform well. Here we are using the RBF kernel, also known as the radial basis function, to build up the SVM model. The two important parameters for RBF kernel are cost and gamma.

We build the model with all the attributes and Figure 3 shows their importance in the SVM model. As demonstrated in Figure 3, the overall accuracy rate is 91.72% and the sensitivity rate is 97.21%.
Neural networks technology mimics the brain’s own problem solving process. Just as humans apply knowledge gained from past experience to new problems, a neural network takes previously solved examples to build a system of “neurons” that makes new decisions, classifications, and forecasts. Neural networks look for patterns in training sets of data, learn these patterns, and develop the ability to correctly classify new patterns or to make forecasts and predictions.

The basic building block of neural networks technology is the single neural processing unit called the neuron. Independent neurons are of little use until they are interconnected in a network of neurons. The network processes a number of inputs from the outside world to produce an output: the network’s classifications or predictions. The neurons are connected by weights, (depicted as lines) which are applied to values passed from one neuron to the next.

As we have done for the other four models, we build a neural network model based on all attributes. As shown in the confusion matrix output below, the overall accuracy rate has decreased to 88.86%, whereas the sensitivity has gone up to 99.99%. This means that the current model is very aggressive and just predicts every client will not subscribe a term deposit. This indicates the model may have some issues and can be further improved.

(5) Neural Network
Figure 2.

Table 7.
Figure 3.

Table 8.

```r
> confusionMatrix(svmpredictions, mytest$y)
Confusion Matrix and Statistics

Reference
Prediction  no  yes
no  11233  755
yes  322   692

Accuracy : 0.9172
95% CI : (0.9123, 0.9218)
No Information Rate : 0.8887
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.5182
Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9721
Specificity : 0.4782
Pos Pred Value : 0.9370
Neg Pred Value : 0.6824
Prevalence : 0.8887
Detection Rate : 0.8639
Detection Prevalence : 0.9220
Balanced Accuracy : 0.7252

'Positive' Class : no
```
MODEL COMPARISON AND DISCUSSIONS

In this research, we have explored five machine learning techniques and built five models to predict if the client will subscribe a term deposit. All of the results are summarized in Table 5. We find that in general, all these five models performed very well and the prediction accuracy rates for DT, RF, and SVM are very close to each other. In terms of sensitivity, NN is the best, whereas NB has the lowest score.

After understanding our data and exploring the several important models to predict if the client will subscribe (yes/no) a term deposit (y), we have arrived at the conclusion that all these models have done a good job and there is no one model that is all the best among them. One limitation of this research is that we do not further improve the models by using some techniques such as attribute selection. This may suggest future research on these five machine learning algorithms is necessary.

Table 9.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Byes</td>
<td>86.54%</td>
<td>89.79%</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>91.79%</td>
<td>96.55%</td>
</tr>
<tr>
<td>Random Forest</td>
<td>91.89%</td>
<td>96.58%</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>91.72%</td>
<td>97.21%</td>
</tr>
<tr>
<td>Neural Network</td>
<td>88.86%</td>
<td>99.99%</td>
</tr>
</tbody>
</table>
REFERENCES


Sipu Hou was a Master’s student in Business Analytics at California State University East Bay. She is enthusiastic about utilizing data to make data-driven decisions and has hands-on experience in data modeling, data-based decision-making, and statistical analysis. She has solid technical knowledge in data mining techniques including dimension reduction, multiple linear regression, KNN, classification and regression trees, random forest, and clustering.

Zongzhen Cai has a background in business analysis and finance analysis. He is a data analyst at Center for Disease Analysis Foundation. He is proficient in data analyst with knowledge of Python, SQL, R and VBA. He is skilled in data mining, machine learning algorithm, statistical analysis of large dataset. He works with the modeling team to analyze data to develop disease modeling strategy and validate model analysis results to support epidemiology team. He has focus on improving the model while developing customized tools for data extraction, transforming and loading, and visualization. He graduated from California State University East Bay with Master of Science in business analytics.

Jiming Wu is a Full Professor and Interim Associate Dean in the College of Business and Economics at California State University East Bay. He received his B.S. from Shanghai Jiao Tong University, M.S. from Texas Tech University, and Ph.D. from the University of Kentucky. His research interests include Big Data analytics, IT adoption and acceptance, and computer and network security. His work has appeared in MIS Quarterly, Journal of the Association for Information Systems, European Journal of Information Systems, Information & Management, Decision Support Systems, and elsewhere.

Hongwei Du is the Chair of the Department of Management, College of Business and Economics at California State University East Bay. He holds a Ph.D. in Operations Research from Florida Institute of Technology, a M.S. in Computer Science from Bowling Green State University and a M.S. in System Engineering from Beijing Institute of Automation. His works have been published in the European Journal of Information Systems, the Journal of Information Technology Theory and Application, the California Journal of Operations Management, the International Journal of Innovation and Learning, the International Journal of Intercultural Information Management, the International Journal of Information and Decision Science, the International Journal of Electronic Healthcare, the Journal of Economic Studies, the Journal of Supply Chain and Operations Management, and Financial Innovation.

Peng Xie is an assistant professor at the College of Business, California State University, East Bay. He received his Ph.D. degree in Business Administration from the Scheller School of Business, Georgia Institute of Technology. His research interest includes social media analytics, cryptocurrency, text mining, and financial technology. His research has appeared in top-quality journals such as Journal of Management Information Systems and Information System Frontiers. He has also been serving as Associate Editor in top IS conferences such as ICIS2021.