UNDERSTANDING THE EFFECTIVENESS OF BANK DIRECT MARKETING

Tarun Gupta, Tong Xia and Diana Lee

1. Introduction

There are two main approaches for companies to promote their products / services: through mass campaigns, which target the general public population, and directed campaign, which targets only a specific group of people. Formal study shows that the efficiency of mass campaign is pretty low. Usually less than 1% of the whole population will have positive response to the mass campaign. In contrast, direct campaign focuses only on a small set of people who are believed to be interested in the product/service being marketed and thus would me much more efficient. Here in our project we focus only on the direct marketing data. Our goal is to predict if a customer will subscribe the service provided by the bank, thereby improving the effect of direct marketing.

2. Data Pre-processing

We are using data from direct marketing campaigns (phone calls) of a Portuguese banking institution. There are 45211 records, which of which is composed of 16 features and 1 binary response. The following table gives a brief introduction of all features.

Name	**Attribute**	**Description**
age	numeric	age of the customer
job	categorical	type of job. e.g. management, entrepreneur, housemaid, student
marital	categorical	marital status
education	categorical	education level. e.g. unknown, secondary, primary, tertiary
default	binary	has credit in default?
balance	numeric	average yearly balance in euros
housing	binary	has housing loan?
loan	binary	has personal loan?
contact	categorical	contact communication type. e.g. telephone, cellular, unknown
day	numeric	last contact day of the month
month	categorical	last contact month of year
duration	numeric	last contact duration in seconds
campaign	numeric	number of contacts performed during this campaign and for this client
pdays	numeric	number of days that passed by after the client was last contacted from a previous campaign, -1 means client was not previously contacted
previous	numeric	number of contacts performed before this campaign and for this client
poutcome	categorical	outcome of previous marketing campaign, e.g. success, failure, other.

The outcome is a binary variable indicating if the client has subscribed a term deposit. The percentage of positive outcomes in the data is 0.11. This gives us a benchmark test error of 0.11 when all outcomes are negative. So, we should focus on recall and precision.

We consider *'contact', 'month' and 'day'* to be non-interesting attributes and ignore them in this analysis.

2.1. Categorical Attributes

Some algorithms like SVM and k-nearest neighbours only accept numerical attributes. Thus, the attributes have to be converted appropriately to numeric values for these algorithms. Binary attributes have been mapped to {no = 0, yes= 1} values. Ordered categorical attributes have been converted to discrete values. For example, '*education'* level has been mapped as {primary = 1, secondary = 2, tertiary = 3} and '*poutcome'* is mapped as {failure = -1, success = +1, other/unknown = 0}. For unordered attributes, the categories have been duplicated to create additional features.

2.2. Feature Scaling

It is important to scale the continuous numeric features to improve data quality. For example, 'balance' attribute contains values like 12686. This can cause multiple. In SVM, this value will blow up for high-dimensional polynomial kernels causing numerical errors in kernel inner products. In regression, the feature matrix may become ill-conditioned or a single feature may dominate other features with small numeric ranges.

3. Model Selection

3.1. Mutual Information

The following list gives attributes in the order of mutual information of a single feature with the outcome:
"balance" "poutcome" "previous" "blueCollar" "single" "management" "technician" "loan" "admin" "divorced" "housing" "services" "married" "retired" "selfEmployed" "entrepreneur" "housemaid" "campaign" "student" "default" "education" "pdays" "duration" "age"

3.2. Stepwise Search Algorithm

We have carried out stepwise model selection in R using AIC which maximizes the likelihood function for the estimated model. Both forward and backward search yield the same model:
y ~ duration + poutcome + housing + job + loan + campaign + education + marital + balance

3.3. Ensemble Classifier

We plan to carry out an ensemble technique like random forests. We can combine the best classifiers to obtain our predictions. In this report, we will combine the classifiers (Logistic Regression, AdaBoost, NaiveBayes and Random Forests) to generate an ensemble to improve the performance across atleast one of the metrics (Test Error, Recall and Precision) for each classifier. The final prediction is made by taking a majority vote among the four classifiers.

3.4. SVM parameter tuning

We need to study the variation as we move the decision boundary using a precision-recall curve. This is accomplished by varying the parameters (C, γ) in the SVM model. The table below shows the results obtained for different parameters using 70% cross-validation. We can observe the precision-recall trade-off in the following results. We also note that the test error increases significantly when we try to improve the recall by using a high value of parameter C. The training error goes down monotonically with increasing values of C and γ.

γ	C	**Training Error**	**Test Error**	**Recall**	**Precision**
0.001	1	0.110	0.115	0.0575	0.7966
0.01	1	0.105	0.1099	0.1877	0.6518
0.1	1	0.0939	0.1069	0.2	0.595
0.1	10	0.0695	0.1087	0.298	0.598
0.1	100	0.0525	0.1207	0.319	0.4985

Cross Validation to find best parameter values for SVM: To find the best tuning parameter C of the SVM learning algorithm, we give a list of candidates for C and γ**.** For each of these candidates, we estimate the 10-fold cross validation error. The optimal C is identified as the one that minimizes the CV error. We then fit the SVM model with the optimal C on the whole training set and use it for future prediction. We use R's tune function to perform cross-validation on a (C, γ) grid using 10% of the data set. The optimal parameter values are obtains as: C = 1, γ = 0.0625. The results seem to change slightly when the tuning is performed again. This can occur because each time the data is partitioned randomly.

4. Results

We randomly choose 70% of the data as the training set, and the remaining as the test set.

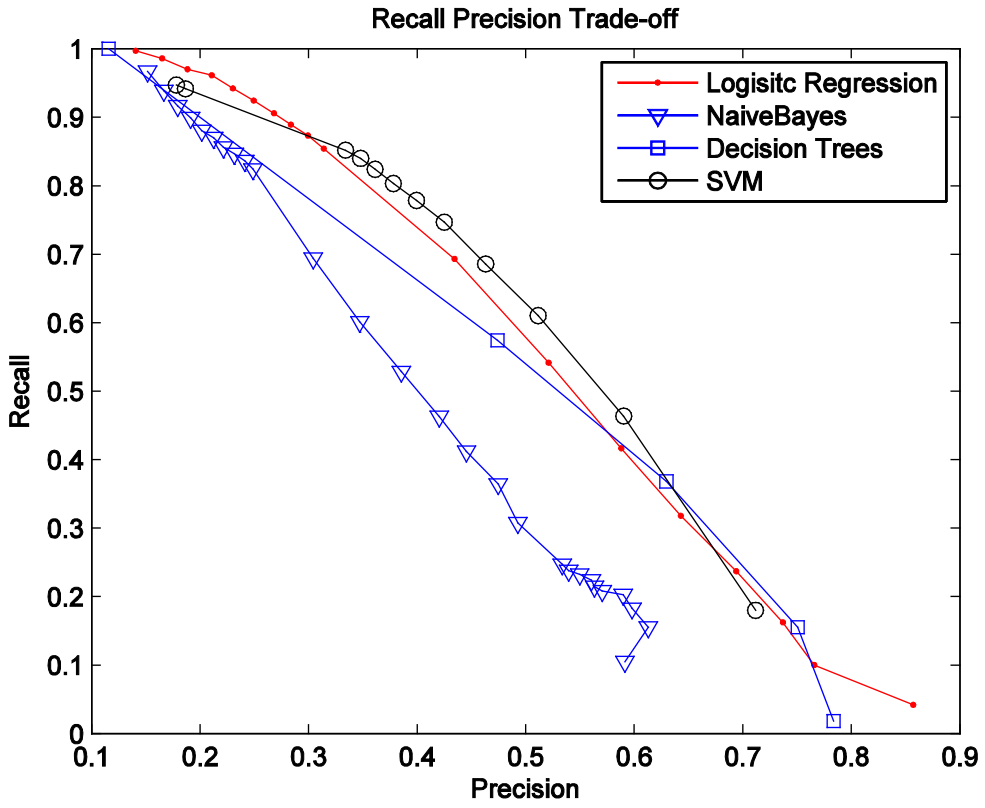
Algorithm	**Training Error**	**Test Error**	**Recall**	**Precision**
Logistic	0.1012	0.0993	0.3170	0.6430
SVM	0.1016	0.1076	0.1798	0.711
NaiveBayes	0.1422	0.1360	0.4630	0.4205
Decision Trees	0.0995	0.0982	0.3681	0.6296
AdaBoost	0.1035	0.1027	0.4394	0.5736
Random Forests (nTrees = 50)	0.0306	0.0983	0.3611	0.6300
Random Forests (nTrees = 500)	0.0305	0.0976	0.3726	0.6324
Random Forests (nTrees = 5000)	0.0296	0.0973	0.3745	0.6343
Ensemble (LR+NB+AB+RF)	**0.07833**	**0.0987**	**0.4420**	**0.5998**

For all the methods except NaiveBayes, the precision is higher than 50%. This is pretty good. Note that the percentage of positive outcomes is 11%. So, compared to mass (random) campaign, our learning procedure is much more efficient.

However, on the other hand, the recall is lower than 50% for all the methods. This means that we are missing more than half of the potential customers. Clearly this result is unacceptable. In the next step, we would focus on how to increase the recall.

Evaluation of Performance Metrics:

We present the recall-precision curve for this application using different classifiers. The curve is obtained by changing the decision boundary threshold in the case of logistic regression, Naïve Bayes and Decision Trees. In the case of SVM, we obtain the curve by assigning different weights to the positive and negative class.



We also observe that for a high recall, low precision setting, the test error goes up significantly due to increase in the number of false positives. In SVM, the low precision (0.33) and high recall (0.85) values correspond to a test error of 22%. But, we are more interested in recall for this application and accept this increase in the number of false predictions to capture most of the potential customers.

5. Conclusions

We have implemented several machine learning algorithms for this application. Although these algorithms fail to be significantly accurate for this application, we are able to obtain high recall classifiers.

SVM outperforms other classifiers in terms of the performance on the recall-precision curve. Logistic Regression seems to work well and narrowly loses compared to SVM.

The ensemble classifier improves the performance in at-least one of the metrics compared to using an individual classifier.