Predicting bank telemarketing call success

1. Introduction

Telemarketing is one of the most cost-efficient forms of advertising, but the conversion rate of telemarketing is dismal. With the Internet of Things introducing other avenues of advertising, as well as Singapore's enactment of the Do Not Call Registry on 14 January 2014 which limits the scope of telemarketing, being able to successfully predict the outcome of a telemarketing call will be crucial to the continued usage of telemarketing.

The objective of this work paper is to develop a ML application which can predict the success of a telemarketing call for a bank's term deposit based on the profile of the client, success of previous telemarketing campaign and economic factors. The ML classification estimators used include Logistic Regression, Support Vector Machine and Tree classifiers.

2. Dataset

A dataset containing 41,188 records of client data from a Portuguese banking institution's telemarketing campaign has been obtained from the following source: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing[1]. Do note that there are 4 datasets, and the dataset with all the features ("bank-additional-full.csv") was used. A copy of the full dataset had been included in the zip file, titled "bank-additional-full.xlsx".

2.1 Preliminary analysis using excel

As the original dataset was using semicolon as the separator, Microsoft Excel (excel) was used to process the dataset using the "Get Data" query to convert the dataset into a csv file for analysis. The processed dataset has been included in the zip file, titled "Processed_bank_full.xlsx".

Based on a preliminary analysis of the processed dataset, the following was noted:

Finding	Action to be taken
- 20 features comprising of both categorical and numeric features, with a known target label	- Categorical features will be encoded via label-encoding or one-hot encoding.
- Categorical features have missing values filled up with the value "unknown" - Numeric features have no known missing values	- To replace "unknown" with NaN in Python to sieve out all columns with missing values, and subsequently replace with the mode of the respective feature
- For feature "pdays", an extreme value of "999" was used to represent that the client was not previously contacted. This may lead to outliers.	- To determine if feature will be used, and if so, replace the extreme value with another value such as (max + 1) or use encoding

[1] Citation of original source:
[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

2.1 Preliminary analysis using excel (cont'd)

Finding	Action to be taken
- Feature naming convention not standardised and not obvious at first glance	- To standardise the naming convention with simple reference names

2.2 Data pre-processing

2.2.1 Data cleansing

Please refer to the python notebook attached in the zip file for the data cleansing.

The following table shows the list of 20 renamed features and the target from the original dataset[2]:

Features	
age	Age of client, in years
job	Type of job, classified based on fixed categories
marital	Marital status (note that widowed are also classified as "divorced")
education	Highest education level attained, based on fixed categories
credit_default	Whether the client has any credit on default
house_loan	Whether the client has a housing loan
personal_loan	Whether the client has a personal loan
contact	Method of contact for this telemarketing campaign
month	Month of last call with client for this telemarketing campaign
day_of_week	Day of last call with client for this telemarketing campaign
duration	Duration of last call with client in this telemarketing campaign, in seconds, feature dropped
num_campaign	Number of contacts performed in this telemarketing campaign; feature dropped
p_days	Number of days from previous telemarketing campaign contact
p_num	Number of contacts performed in previous telemarketing campaign
p_outcome	Outcome of previous telemarketing campaign
EVI	Quarterly employment variation index of Portugal
CPI	Monthly consumer price index of Portugal
CCI	Monthly consumer confidence index of Portugal
euribor3m	Daily 3 month Euro Interbank Offered Rate
country_employment	Quarterly average of total number of employed citizens, in thousands
Target	
outcome	Whether this telemarketing call is successful

Dropped features

The features "duration" and "num_campaign" are features obtained after making the telemarketing calls in the current telemarketing campaign. As such, they were dropped to train a more realistic model.

[2] Description of features extracted from the file "bank-additional-names.txt" (attached in the zip file) and [Moro et al., 2014].

2.2.1 Data cleansing (cont’d)

Duplicates

Due to the absence of a unique identifier such as client ID in the dataset, all instances (rows) were treated as unique and was not be dropped although there were instances with the same values. This mirrors real-world data where different clients can have the same attributes.

2.2.2 Feature engineering

Debt

There are 3 features for debt in the original dataset, namely “credit_default”, “house_loan” and “personal_loan”. As these 3 features are similar in nature, they had been aggregated into a new feature “debt” to describe if the client has any debt. The purpose of this new feature is to determine if the presence of debt has any bearing as to whether the client will take up a term deposit.

Features	
debt (new feature)	Whether the client has any outstanding debt from credit default, housing loan or personal loan

Previous campaign outcome

There are 3 features relating to the previous telemarketing campaign, namely “p_days”, “p_num” and “p_outcome”. The focus of these 3 features should be on whether the success or failure of a previous telemarketing campaign can be used to predict the outcome of the next telemarketing call. As such, features “p_days” and “p_num” were dropped.

Employment indicators

There are 2 features relating to employment indicators, namely “EVI” and “country_employment”. The feature “EVI” is a better measure of employment since it shows the change in employment on a quarterly basis. For instance, a positive EVI would indicate an increase in quarterly employment of Portugal, and vice versa. However, the feature “country_employment” is just a static figure of the number of employed individuals in that quarter with no comparative component. Hence, the feature “country_employment” was dropped.

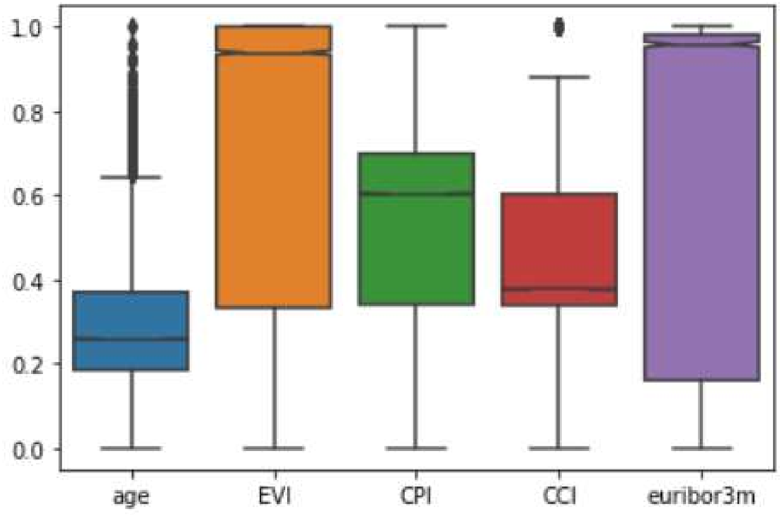
2.3 Data visualisation

The following visualisation plots were used to analyse the features in python:

- Boxplot to identify outliers
- Heatmap to identify feature correlation

2.3 Data visualisation (cont'd)

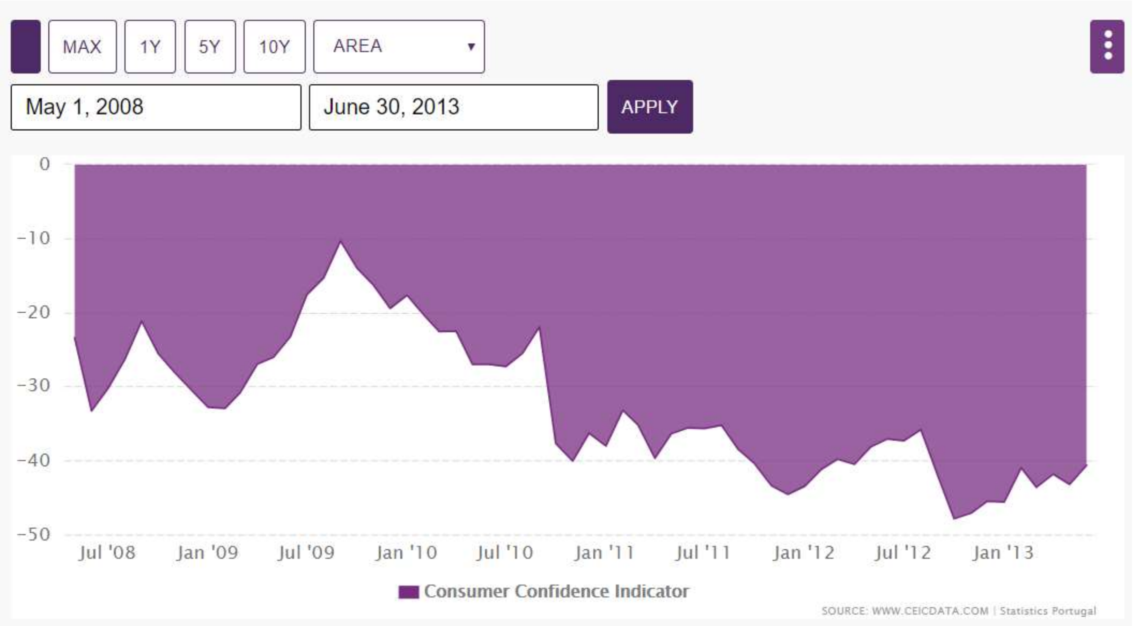
<u>Boxplot of numeric features</u>



Based on the boxplot of numeric features, high outliers were noted for features "age" and "CCI".

For feature "age", these high outliers are not unusual as the number of people who live past the mean life expectancy decrease exponentially as age increases. Hence, outliers for feature "age" was not removed.

For feature "CCI", based on the original research performed, the dataset was extracted from May 2008 to June 2013 [Moro et al., 2014]. Based on an analysis of Portugal's CCI during this period (refer to the graph below), it was noted that the highest CCI was approximately -10 during this period, which is higher than the max of the feature "CCI" (-26.9). As such, it is unlikely that the high outliers were erroneous. Moreover, these outliers all seem to lie within a specific time period, given that they have approximately the same value. Hence, outliers for feature "CCI" was not removed.



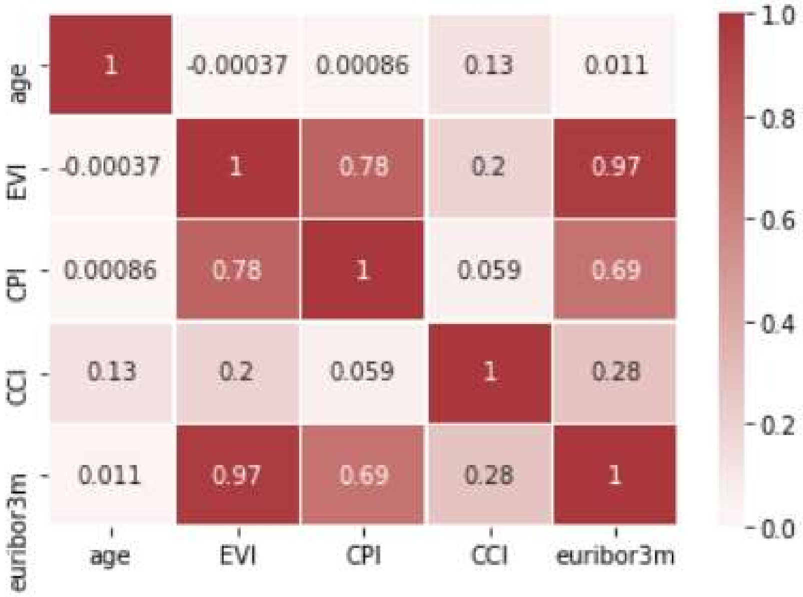
May 1, 2008

June 30, 2013

SOURCE: WWW.CEICDATA.COM | Statistics Portugal

2.3 Data visualisation (cont'd)

Heatmap correlation of numeric features



Based on the heatmap of numeric features, a strong correlation of 0.97 was noted between features "EVI" and "euribor3m". Considering that the feature "EVI" is a better feature since it directly pertains to Portugal whereas the feature "euribor3m" pertains to the Eurozone market which Portugal is a part of, the feature "euribor3m" was dropped.

Features list after pre-processing

The list of 12 features after data pre-processing is as follows:

- age
- job
- marital
- education
- contact
- month
- day_of_week
- p_outcome
- EVI
- CPI
- CCI
- debt
- outcome (Target)

Encoding

Categorical features and target have been transformed via One-Hot Encoding and Label Encoding, respectively.

2.3 Data visualisation (cont'd)

Train-validation-test split

The dataset was split into a training set and testing set using a ratio of 80% and 20% respectively. Stratified split was used due to the highly imbalanced dataset (only 11.3% "Yes" outcome). The training set was further split into training and validation set using a ratio of 80% and 20% respectively. The validation set will be used to tune and select the best model, before finally testing the selected model with the test set. All numeric features in train set were scaled using MinMaxScaler.

3. Model training and testing

3.1 Model selection

The following classifiers were selected for the training set:

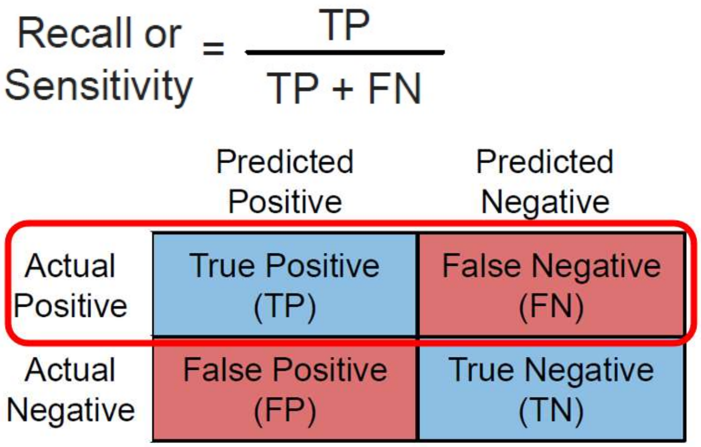
- Tree classifiers
- Logistic Regression
- Support Vector Machine

Certain hyperparameters for each model will be varied to determine the best model for each classifier. Subsequently, the best model from each classifier will be used on the validation set to determine the overall best model, and the selected model will then be used on the test set for evaluation.

Performance Indicator

Due to the high data imbalance, accuracy is not good indicator of performance. Recall was adopted as the performance measure instead to reduce the occurrences of False Negatives (predicting a failed telemarketing call when it will be successful).

False Negatives need to be minimised as telemarketing calls would only be made based on a predicted positive outcome of the prediction (prediction of successful call). Moreover, the number of actual successful calls is low (as seen from the highly imbalanced dataset), hence each actual successful call needs to be captured as a True Positive by having a higher Recall score (minimise False Negatives).



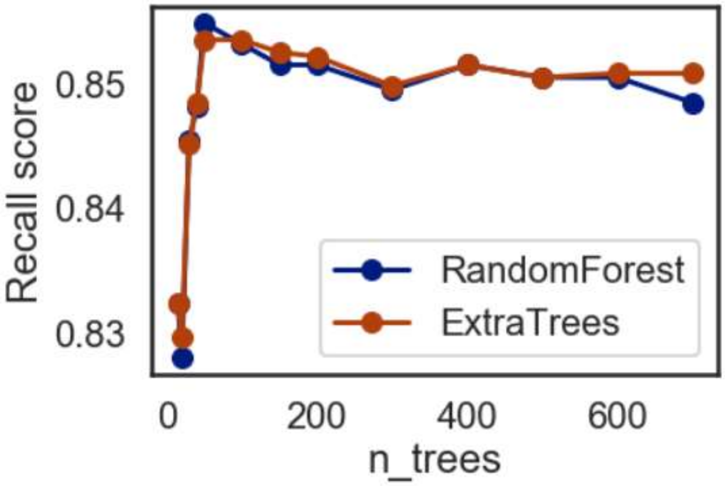
$$\text{Recall or Sensitivity} = \frac{TP}{TP + FN}$$

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

3.1 Model selection (cont'd)

Tree classifiers

Due to the desire for more randomness, the tree classifiers Random Forest classifier and Extra Trees classifier were used.



As seen above, based on a comparison of the Recall scores using a varying number of estimators, the Random Forest classifier using 50 estimators provided the highest Recall.

Logistic Regression CV

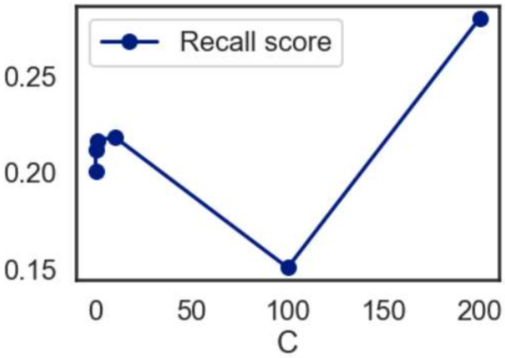
For the logistic regression estimator, to prevent over-fitting, Elastic Net regularization was used, in conjunction with cross-validation using the default value of 5-folds. The mean score of each Elastic Net regularization ratio of L1 (Lasso regression) and L2 (Ridge regression) was computed as shown below. The best mean score using Recall was when L1 = 0 i.e. using only Ridge Regression for regularisation.

L1 ratio, Lasso	L2 ratio, Ridge (1 – L1 ratio)	Mean Recall
0.0	1.0	0.168316
0.1	0.9	0.166835
0.2	0.8	0.165926
0.3	0.7	0.165455
0.4	0.6	0.165051
0.5	0.5	0.165152
0.6	0.4	0.165286
0.7	0.3	0.165455
0.8	0.2	0.165758
0.9	0.1	0.165825
1.0	0.0	0.166061

3.1 Model selection (cont'd)

Support Vector Machine

The linear support vector classifier was used, and the regularization parameter "C" varied to identify the best performing regularization parameter using Recall as the performance metrics. The best Recall was noted using C = 10. The sharp rise in Recall after C = 100 is due to overfitting.



3.2 Testing on validation set

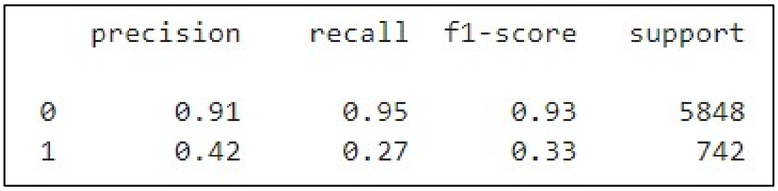
The best models from each of the 3 estimators were used on the validation data. The performance metrics of each model is summarized below.

Model	Accuracy	Precision	Recall	F1	AUC
Random Forest	0.876328	0.421842	0.265499	0.325889	0.609665
Logistic Regression CV	0.897269	0.640693	0.199461	0.304214	0.592634
Linear Support Vector	0.896662	0.638009	0.190027	0.292835	0.588174

Random Forest classifier had the best score in terms of Recall.

However, the performance scores for all models are noticeably low, especially for Recall. This is due to the highly imbalanced dataset that was used to train the model.

For instance, as seen in the performance score of the different target labels below for the Random Forest model, label "0" ("no" outcome) had significantly higher performance scores across all categories compared to label "1" ("yes" outcome). Label "0" was also observed to have significantly higher count (6.9 times higher) than label "1".



```
     precision    recall  f1-score   support

0         0.91      0.95      0.93      5848
1         0.42      0.27      0.33       742
```

To reduce the influence of the imbalanced dataset, two methods were considered:

- Synthetic Minority Oversampling Technique for Nominal and Continuous (SMOTE-NC)
- Assignment of class weights to target labels

3.3 Model tuning

3.3.1 Addressing imbalanced dataset

SMOTE-NC

SMOTE-NC generates synthetic data of the minority label ("yes" outcome) such that the number of data for each label is the same. The synthetic data is generated based on k-nearest neighbours of the minority label using Euclidean distance of both continuous and categorical features.

For the train data, number of synthetic data generated for the minority label is 20420, resulting in the majority and minority labels having the same data counts of 23390. This thus resolves the imbalanced data through synthetic data generation.

Model	Accuracy	Precision	Recall	F1	AUC
Random Forest	0.856146	0.348083	0.318059	0.332394	0.621239
Logistic Regression CV	0.817602	0.332362	0.614555	0.43141	0.72896
Linear Support Vector	0.815933	0.330209	0.617251	0.430249	0.729196

As seen in the table above, using SMOTE-NC method had improved the Recall for all models, although Accuracy fell.

Class Weights

Class weights were determined based on the number of occurrences of each outcome in the training set only to prevent data leakage. Higher weight will be placed on the class with a lower occurrence.

Model	Accuracy	Precision	Recall	F1	AUC
Random Forest	0.868589	0.385609	0.281671	0.325545	0.612364
Logistic Regression CV	0.818058	0.333333	0.615903	0.43256	0.729805
Linear Support Vector	0.818361	0.334545	0.619946	0.434577	0.731741

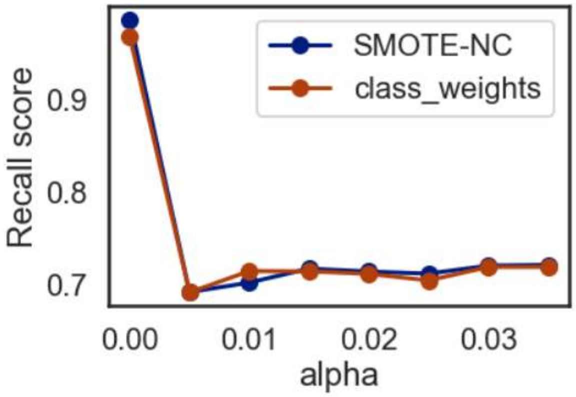
As seen in the table above, using class weights produced similar, though slightly higher performance scores as compared to SMOTE-NC.

Random Forest pruning

The Random Forest model had the worst performing Recall despite the use of imbalanced methods SMOTE-NC or class weights. This is due to severe over-fitting onto the train dataset. To mitigate this, minimal cost-complexity pruning was implemented over a range of alphas to determine the best alpha using the training set.

3.3.1 Addressing imbalanced dataset (cont'd)

Random Forest pruning (cont'd)



As seen above, at alpha = 0, Recall was almost perfect (close to 1), but when alpha increased, Recall fell to approximately 0.7, which is in line with the other models. Recall score stabilised around alpha = 0.03 for both SMOTE-NC and class weights methods. As SMOTE-NC method consistently resulted in higher Recall, SMOTE-NC was adopted for the Random Forest model.

The pruned Random Forest model at alpha = 0.03 was then retrained and tested with the validation set. The final performance scores of the 3 models are as follows:

Model	Method	Accuracy	Precision	Recall	F1	AUC
Random Forest	SMOTE-NC	0.721093	0.241753	0.691375	0.35824	0.708119
Logistic Regression CV	Class weights	0.818058	0.333333	0.615903	0.43256	0.729805
Linear Support Vector	Class weights	0.818361	0.334545	0.619946	0.434577	0.731741

Based on the results for the 3 estimators, the Random Forest model using SMOTE-NC resulted in the highest Recall, though with a lower Accuracy (Accuracy = 0.721) as compared to the accuracy of a blind prediction that all outcomes are "No" (Accuracy = 0.887).

However, since Recall is the main performance indicator, the Random Forest model was selected to predict the outcome of a telemarketing call.

3.3.2 Feature importance

Based on the feature importance extracted from the fitted Random Forest model, the top 5 features of importance in descending order is as follows:

Feature	Coefficient
EVI	0.310558
p_outcome_success	0.181033
p_outcome_nonexistent	0.162162
CCI	0.134271
contact_cellular	0.108108

3.3.2 Feature selection (cont'd)

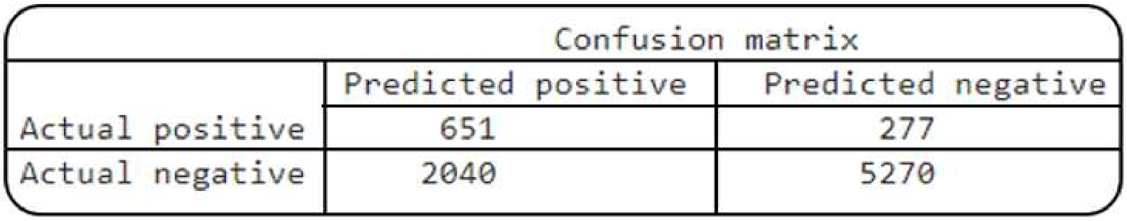
As all the features were scaled using MinMaxScaler during data pre-processing, all feature values only range from 0 to 1, hence the features are directly comparable via their coefficient.

It can be inferred that the economic indicators (EVI, CPI and CCI) all played a significant role in determining the outcome of a telemarketing call, with all 3 in the top 7 most important features.

Moreover, it is noted that the categorical features "job", "marital", "education", "debt", which were determined to have missing values during data pre-processing, all had low importance (< 0.005), with some feature values even having no importance (importance = 0). Hence, these features could be dropped instead of filling the missing values with the mode during data pre-processing.

4. Evaluation

The selected model (pruned Random Forest model with class weights) was used on the test set.



Confusion matrix

	Predicted positive	Predicted negative
Actual positive	651	277
Actual negative	2040	5270

On the assumption that telemarketing calls are only made when the prediction is a positive outcome (predicted successful call):

Total calls made with prediction (True Positive + False Positive) = 651 + 2040 = 2691

Total calls made without prediction (total test data) = 651 + 277 + 2040 + 5270 = 8238

% reduction in total calls made = (8238 – 2691) / 8238 = **67.3%**

% calls that were successful with prediction = 651 / 2691 = 24.2%

% calls that were successful without prediction = (651 + 277) / 8238 = 11.3%

% increment/(decrement) in successful calls with prediction = (24.2% - 11.3%) / 11.3% = **115%**

As can be seen, the use of the pruned Random Forest model resulted in a 67.3% reduction in total calls made, while having a 115% higher rate of successful calls. Clearly, this represents huge cost savings in terms of manpower and time, given that traditionally, telemarketing has a low rate of success (11.3% based on this dataset) while being labour-intensive. This will free up resources for the bank to work on other more lucrative avenues.

The trade-off to using the pruned Random Forest model is the lower overall number of successful calls (29.8% lower).

4. Evaluation (cont’d)

In the original research paper in Moro et al., 2014, other features not in this dataset, such as bank profiling indicators, experience of agent making the telemarketing call and the National monthly average of deposit interest rate, were found to have significant influences on the AUC. Hence, if the dataset provided was as comprehensive, perhaps the Recall and Accuracy of the pruned Random Forest model could be improved and reduce the number of False Negatives.

5. Conclusion

The pruned Random Forest model successfully met the objective to predict the success of a telemarketing call for a bank’s term deposit.