Predict Term Deposit Subscription For Target Marketing

Jake Warren
contact.jakewarren@gmail.com

Abstract

A term deposit is a fixed-term investment that includes the deposit of money into an account at a financial institution. Term deposit investments usually carry short-term maturities ranging from one month to a few years and will have varying levels of required minimum deposits. A customer will deposit or invest in one of these accounts, agreeing not to withdraw their funds for a fixed period in return for a higher rate of interest paid on the account. If a customer places money in a term deposit, the bank can invest the money in other financial products that pay a higher rate of return (RoR) than what the bank is paying the customer for the use of their funds. The bank can also lend the money out to its other clients, thereby receiving a higher interest rate from the borrowers as compared to what the bank is paying in interest for the term deposit. Net interest income is the amount of money earnt from the offset of the interest from clients borrowing the term deposit money and the interest given to clients for subscribing to a term deposit.

I. Introduction

Target marketing is vital for many businesses that offer products to their customers, to be able to classify whether a customer will or will not respond to an offer has major advantages when it comes to profitability and expenditures. Classification for target marketing can also have its problems; class distribution for responders/non-responders is usually extremely imbalanced, 1 in 10 or 1 in 100 can be quite common ratios for responders. This poses a problem when using accuracy as a metric because a model could just predict the majority class every time and still gain high accuracy but be useless in practice when the minority class is of most importance. For this classification problem the F-beta metric proves to be a better choice as we can provide more weight towards the goals we want our model to achieve.
Our aim with this project is to maximise profits by targeting customers with the highest probability of subscribing to a term deposit. We will use the F-beta metric (weighted harmonic mean of precision and recall) with a weighting of 1.5 which is weighted more towards recall (reducing false negatives) but with some weighting towards precision (reducing false positives) aswell. Having F-beta weighted this way allows us to gain more in profit by not missing potential subscribers but with a slight trade off in gaining some extra costs, since our profit margin is quite big this allows our model to extract more profit than if we weighted F-beta more towards reducing costs.

Project outline:
For this project we chose an arbitrary number of £10000 for the term deposit amount for 2 years earning the client 5% interest. We will assume we will gain 10% interest over the 2 years from borrowers leaving us with 5% profit, which amounts to £500 in net interest income per term deposit subscription over the 2 year term, there will be a cost per targeted client of £30. Another assumption that we have made is that the distribution of the target variable classes are the same as the population, this is so we have class priors for the expected profit calculation. We found that if we target everyone on average we will achieve a profit of £25 per client, with our final model we can achieve on average a profit of approximately £33 per client if we target the top 25% of clients with the highest probability of subscription (modeled for p(S|x), where S is subscription and x is the client), this is a 32% profit increase over the base rate £25.

II. Dataset

The data were obtained online and is open to the public for machine learning purposes, the data collected has 21 features and 41188 observations containing customer information, see table 1 below for description of the data. Of the 21 features 11 are categorical including the target variable and are comprised of nominal, ordinal, dichotomous and cyclical data, the remaining 10 features are continuous. The target variable ‘y’ describes whether a customer subscribes to a term deposit or not and therefore consists of either yes or no, this variable is highly imbalanced with 32885 no’s and 4176 yes’s. As this is a relatively small dataset we opted for a ten percent testing split leaving us with more data to train on, this translates to 37069 data observations for training and 4119 for holdout/testing.

Table 1. Data Description

Feature	Date Type	Description
Job	Nominal	Occupation of client
Marital	Nominal	Clients marital status
Education	Ordinal	Clients education
Default	Nominal	Client has credit in default
Housing	Nominal	Client has housing loan
Loan	Nominal	Client has personal loan
Contact	Dichotomous	Contact communication type
Month	Cyclical	Last contact month of year
Day_of_week	Cyclical	Last contact day of the week
Poutcome	Nominal	Outcome of previous marketing campaign
Age	Continuous	Clients age
Campaign	Continuous	Number of contacts performed during campaign
Pdays	Continuous	Number of days that passed by after the client was last contacted from a previous campaign
Previous	Continuous	number of contacts performed before this campaign and for this client
emp.var.rate	Continuous	employment variation rate - quarterly indicator
cons.price.idx	Continuous	consumer price index - monthly indicator
cons.conf.idx	Continuous	consumer confidence index - monthly indicator
euribor3m	Continuous	euribor 3 month rate - daily indicator
nr.employed	Continuous	number of employees - quarterly Indicator
y	Dichotomous	has the client subscribed a term deposit

III. Features // Preprocessing

During data cleansing we removed duplicated entries and found features that had a number of values as unknown, this can be assumed as information that the bank doesn't have for the customer. There are a few ways we could deal with these values; *i.* imputation where we compute values for the unknown values, for example the mean/median value of the feature or prediction of value, *ii.* keep them as their own category, *iii.* removal of data entry. For feature/target analysis we used a number of correlation/association techniques including; spearman rank, cramer's v, phi and kendall tau coefficients, logistic/linear regression and information gain. We found the relationship between the features with unknown values and the target variable were relatively weak so these features were removed. Figures 4 and 5 show the information gain and correlation between features and target, the features that had the strongest relationship with the target were; age, contact, campaign, pdays, previous, poutcome, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr.employed. Subsequently we removed all features that had weak to no relationship with the target. During analysis we noticed we had multicollinearity within the data which led us to believe that tree induction models or dimensionality reduction techniques would provide the most optimal outcomes. We cycled through feature extraction/creation finding features with a good to strong relationship were; age_bin - binning age in 10's, log_campaign and log_pdays - because of exponential distributions for campaign and pdays a log of power 10 transformation was used, pdays_contact - was client previously contacted, season - seasonality extracted from month then a sin and cosine transformation was applied to both season and month to extract their cyclical nature. Due to the variability of the feature values we scaled quantitative features so that they have a mean of 0 and deviation of 1 and qualitative features were one hot encoded.

We also opted to use a second set of the same data but using principle components analysis to reduce the dimensions of the data using linear combinations of the original features. Figure 3 shows we can retain 98% variance of the data by only extracting 12 components/features from applying PCA.

fig 3. Feature extraction using PCA

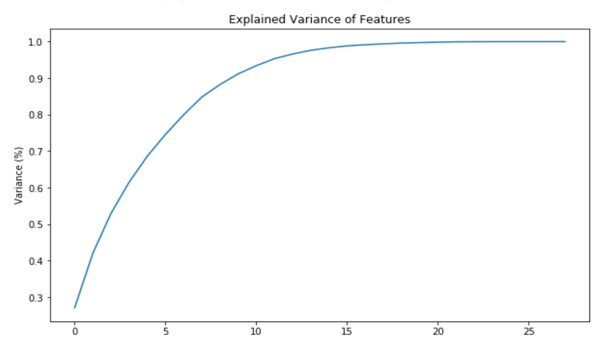


fig 4. Feature information gain

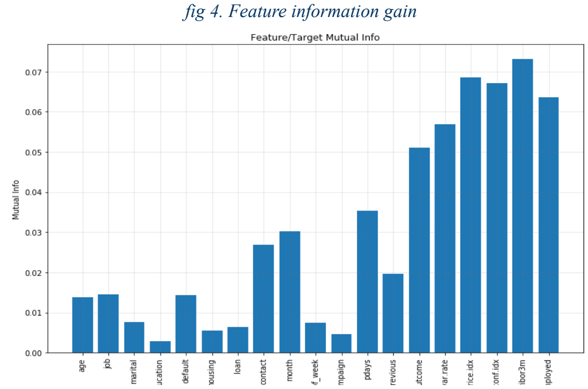
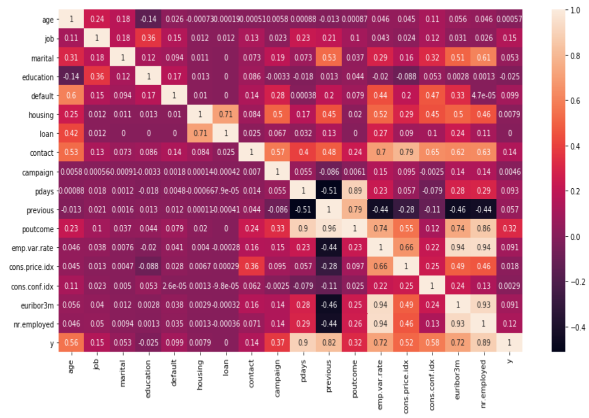


fig 5. Feature correlation/association heatmap



IV. Models

Since this is a classification task supervised learning algorithms were used and before testing began we acquired a base rate metric and a simple model (best feature only) metric as a base for other models to improve on. The base rate F-beta score we wanted to improve on was 0.292 and the base rate profit achieved was £25, the simple model achieved an average F-beta score of 0.4463 and the simple model achieved a profit of £30.53. Since we had found outliers in the data that were important to keep we expected that tree based algorithms would perform the best as they are more robust to outliers but we tested a number of other model algorithms as well as there is no right model for the data, these models were; logistic regression, support vector classification, k-nearest neighbours, decision tree, random forest, gradient boosting, adaboost, gaussian and bernoulli naive bayes . As we have a severe imbalance with the target classes a number of data balancing techniques were implemented to help improve the performance of our models, these techniques include oversampling (synthetically creating minority class data), undersampling (removing data from the majority class) and a combination of both techniques. The undersampling algorithms implemented were; ***i.*** tomek links - remove majority examples at the class borderline that have minority examples as nearest neighbours, ***ii.*** edited nearest neighbours - removes misclassified observations, ***iii.*** repeated edited nearest neighbours - repeatedly applies edited nearest neighbours until no more observations can be removed, ***iv.*** one sided selection - a combination of applying tomek links then applying condensed nearest neighbor method to remove majority examples far away from the class borderline, ***v.*** neighbourhood cleaning rule - combines the condensed nearest neighbor method to remove redundant majority examples and edited nearest neighbor method to remove misclassified majority examples and ***vi.*** instance hardness threshold - removes overlapping class observations. The oversampling algorithms implemented were; ***i.*** smote - selects minority examples close to each other and creates new examples between them, ***ii.*** borderline smote - creates observations from the misclassified observations nearest the class border, ***iii.*** svmsmote - creates minority observations from misclassified minority observations close to the class decision boundary obtained by the svm algorithm and ***iv.*** adaptive synthetic sampling - generates more minority observations in the feature space where a low number of minority observations are found and generates less or no observations where a high number of minority observations are found. A combination of oversampling and undersampling was used using smoteenn which combines smote that selects minority examples that are close to each other and creates a new example between the minority examples and edited nearest neighbours to remove misclassified examples from the majority class.

Once we found the top performing models and their data balancing techniques, a tailored feature selection was performed for each model using the technique sequential forward feature selection. This will sequentially select features that improve the F-beta score and remove features that do not for each model helping to achieve the best F-beta scores possible. We implemented GridSearchCV which is a cross validation technique applied to the parameters of a model to fine tune the models and enhance their performance. All tests were performed using 5-fold cross validation, this way we can see reliability and deviation of each model and gain the benefit of performing multiple tests on the training data, all model metrics obtained were compared with holdout data test metrics to insure generalisation on unseen/new data.

V. Results

The models were tested with the different sampling techniques above and were applied to the data as it was and also on the PCA reduced data. Tables 2 & 3 show the top outcomes for each model for both tests. The models that were chosen to explore further were; gradient boosting with svms oversampling, adaboost with smote oversampling and svc with svms oversampling on pca reduced data, we also opted to test a voting classifier using the gradient boosting and adaboost models. Once we had our top performing models we then obtained the best features for each model using sequential forward feature selection as each model can rank the features importance differently.

From table 4 we can see that all models out performed the base rate model (Fbeta - 0.292) and the simple logistic regression model that we implemented (SM LR), the voting classifier (VC(GBAB)) returned the best performance with a test fbeta score of 0.538 and a roc auc score of 0.752, the test fbeta score is approximately a 16 percent increase over the simple model test score.

Table 2. Model performance

Model	Sampling Model	Fbeta	Std
Gradient boosting	OS - SVMS	0.518468	0.014760
Adaboost	OS - SMOTE	0.509805	0.0155343
Logistic Regression	OS - SVMS	0.507214	0.015340
K-NN	US - NCR	0.476729	0.016910
Random Forest	US - NCR	0.470034	0.010784
SVC	OS - SVMS	0.446975	0.012039
Decision Tree	US - NCR	0.445706	0.011067
Gaussian NB	OS - Borderline SMOTE	0.439344	0.014009
Bernoulli NB	OS - Borderline SMOTE	0.413997	0.010134

Table 3. Model performance with PCA reduction

Model	Sampling Model	Fbeta	Std
SVC	OS - SVMS	0.518635	0.013740
Gradient Boosting	OS - SVMS	0.517086	0.016384
Adaboost	US - IHT	0.498121	0.014048
K-NN	US - IHT	0.477500	0.013253
Logistic Regression	OS - SVMS	0.472734	0.017050
Gaussian NB	OS - Borderline SMOTE	0.456591	0.00373
Random Forest	US - IHT	0.438395	0.009065
Bernoulli NB	OS - SMOTE	0.431444	0.013737
Decision Tree	US - SMOTEENN	0.418496	0.012075

Table 4. Final model metrics

Model	Avg Fbeta	Std	Test Fbeta	AUC
VC(GBAB)	0.531324	0.012398	0.538495	0.752837
SVC	0.522431	0.013640	0.534615	0.749227
Gradient Boosting	0.528438	0.013163	0.529070	0.747552
Adaboost	0.512928	0.014486	0.526672	0.746851
SM LR	0.446351	0.011775	0.465065	0.730932

Figures 4, 5, 6 and 7 below show plots for the final test of the models. Figure 4 shows the profit curves of each model; profit against percentage of data ranked by probability of subscribing, as can be seen the models perform similarly but with the voting classifier as top performer peaking at approximately 33 at around 25% of the data. Figure 5 shows the cumulative response curve; percentage of true positives against percentage of data, the gradient boosting, adaboost and voting classifier models all produce similar ratios of true positives as the percentage of data increases, at 20% of the data they all perform 3 times better than random. Adaboost has a slight increase in true positives around 40-70% of the data, the simple and svc model have a slightly worse ratio with some dips in true positives as more of the data is predicted. All models perform better than random.

Figure 6 shows ROC curves; percentage of true positives against the percentage of false positives. This graph is very similar to the cumulative response graph as they both include the true positive rate but we can see the percentage of false positives predicted as well. Again the gradient boosting, adaboost and voting classifier perform best.

Figure 7 shows the lift curves of each model and we can see that the voting classifier and adaboost models have around a 7 times lift over random at just under 5% of the data. All models produce a similar lift from 10% of the data onwards.

fig 4. Model profit curves

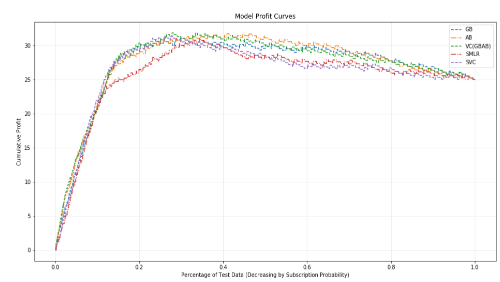


fig 5. Model cumulative response curves

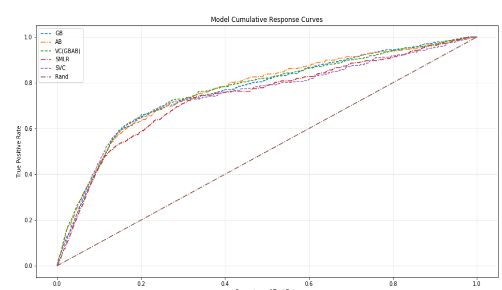


fig 6. Model ROC curves

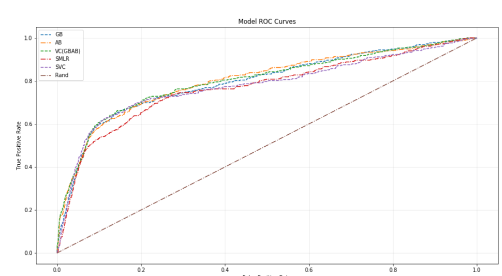
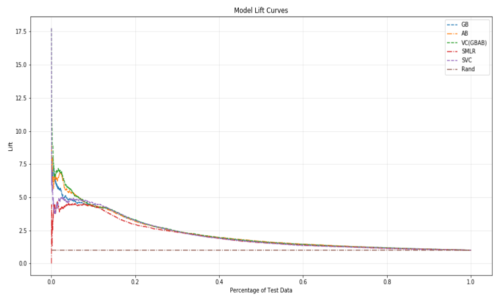


fig 7. Model lift curves



VI. Conclusions

The goal of this project was to not only predict whether a client would subscribe to a term deposit but to also increase profitability for the bank, this way the bank can save time and money by not targeting clients that will not subscribe to a term deposit and increase profits by targeting those clients that do. We found which features correlate/associate best with subscribing to a term deposit; log_pdays, previous, poutcome, emp.var.rate, euribor3m, nr.employed, month and quarter. From the analysis of the test results the two models that show the best performance are adaboost and voting classifier, if we just want to predict subscription or no subscription then either of these models would be top choice but for profitability the voting classifier producing a 32% increase in profits by targeting the top 25% of clients over the base rate of targeting everyone (£33 on average per client over £25 on average per client) this would be our proposed final model. For future use with the voting classifier model we would estimate client subscription probabilities, rank them most likely to least likely to subscribe and target the top 25% of clients to gain the most profit.

VII. Future Improvement

Although having successfully trained various models that out perform a baseline and simple model whilst also proving to be profitable on future data there are a couple of procedures that could be implemented to improve future model induction; ***i***. During analysis of the data we noticed a number of features with unknown values, although these features showed minimal correlation/association with the target if we applied an imputation technique such as computing similarity from nearest neighbours we might find that some of the features do correlate/associate with the target and therefore improve model performance.

ii. As this is only a relatively small dataset if we were able to obtain more data it is possible we could gain further insights and improve on model performance. ***iii.*** During exploratory data analysis we could apply a clustering algorithm to see if the data form into any natural clusters highlighting different client demographics i.e. clients married and 40+ are more likely to subscribe to a term deposit.