Evaluation of Bank Marketing Models

# Introduction

Marketing campaigns are a common practice within the banking industry. This research uses a dataset provided from the UCI Machine Learning repository [23] about a Portuguese bank. This is basically a binary classification problem. The purpose of this research paper is to provide an effective telemarketing strategy to sell term deposits to the bank’s clients.

Classification models (such as Decision Tree, Naïve Bayes, Random Forest, K-Nearest Neighbour, Support Vector Machine, Neural Network, and Logistic Regression) are used to predict the outcome that a bank customer will accept a term deposit. The performance evaluation of these algorithms will be verified using ROC (Receiver Operating Characteristics) curve and/or AUC (Area Under the Curve).

# Literature Review

We wanted to review literature that discussed models that were used to solve a binary classification problem (in particular, bank data), and the methods used to compare them. Then, other secondary issues included what models or processes were used to select features (dimensionality reduction) before using the models, how was the imbalanced data handled (since our data has 87%/13% ratio), and how was the data transformed from numeric to categorical and vice/versa, and finally how was the data ‘normalized’ or scaled?

The initial dataset was provided by Moro *et al.* [1], and they have written a number of papers on the Portuguese bank data, such as [1-4]. In paper [1] Moro *et al.* discussed how to use data mining techniques (the Cross-Industry Standard Process for Data Mining or “CRISP”) when formulating models that could explain the success of a contact i.e. if the client subscribes to the deposit. This article is one of the earliest ones on the Portuguese bank marketing dataset. The data used was larger, and had more outcomes, than the dataset we are using. Data was discarded that had missing values (we had none) and by reviewing the data alone, about half of the attributes were deleted. Three different models were used (Naïve Bayes, Decision Trees and Support Vector Machines). All three models were tested and validated with a holdout split of a training set (2/3) and test set (1/3). Twenty iterations were run for each model. ROC curves and LIFT analysis were performed to compare the models. The results were good, with a minimum AUC of 87%. However, rebalancing, normalizing, discretizing, outliers and “unknown” categories were not discussed.

In paper [2] Moro *et al.* use the same dataset with several classification models (Logistic Regression (LR), decision trees (DTs), neural networks (NNs), and support vector machines (SVMs). It is well known that the LR and DT provide an understandable model while providing good predictions whereas NNs and SVMs are hard to understand but are much more flexible. The dataset also included economic data which our dataset does not. Imbalanced data is discussed but left as is. Feature selection was done in two steps. In the first step, business experts were used to filter down the attributes to 22, then an adapted forward selection method was used. The results were compared for the 4 models using AUC and ALIFT and the results were good for all four models. The minimum time for feature selection to run one of the methods was 53 hours. We may not be able to run similar methods or models with a MacBook Air. For the NN model, a highlight is a Table of the relative importance of the 22 attributes, along with an extracted decision tree from the NN model.

In paper [5] four classification models are used on the Portuguese bank data to determine which models work well, and what factors are the most important. The models used are multilayer perception neural network (MLPNN), augmented Naïve Bayes (TAN) or Bayesian Networks, logistic regression, and decision tree model C5.0. The dataset seems to be left it as is as there is no more discussion on modifying the data. Further, the author did not discuss rebalancing, normalizing, discretizing, or what to do with any outliers and “unknown” categories. The author used classification accuracy, sensitivity and specificity to measure the performance of the models. The models performed well, with a minimum accuracy of 89.16% on the training set and 88.75% on the testing set for TAN, and the other 3 models performed even better.

In paper [6], the same Portuguese bank data is used with models in a WEKA package. The two classification models used are Bayes Net algorithm and Naïve Bayes algorithm. The dataset attributes are discussed but then the discussion moves on quickly to the models. 10-fold cross validation is used for the training and test sets. The authors did not discuss rebalancing, normalizing, discretizing, or what to do with any outliers and “unknown” categories. Further, there was no discussion on what attributes worked the best. Although the authors use WEKA to run the models, it was useful to know what models were actually used. The evaluation metrics were the Confusion Matrix, Kappa, TP Rate, FP Rate, Recall, Precision, F-Measure and ROC Area. Both models performed well with Bayes Net having 81.5% accuracy and Naïve Bayes with 78% accuracy.

Feature selection (dimensionality reduction) is covered in [7-9]. In particular, paper [7] Ladyzynski *et al.* discusses dimensionality reduction in that it can fall into two categories: feature extraction (i.e. Principal Component Analysis) and feature selection. Feature selection is then used by selecting a subset of the original features and an algorithm that does this is called ‘Boratu’. In using Boratu, the number of attributes decreased from 616 to 164. The purpose was to identify models for identifying customers interested in credit products for a large bank in Poland. Three machine learning algorithms were used: 1) classification trees - CART, 2) Random forests and 3) Deep belief networks – DBN. We intend to try Boratu on our data.

It is well-known that there are four type of methods used to rebalance the data when the minority class is overwhelmed by the majority class: 1) undersamping i.e. EasyEnsemble algorithm; 2) oversampling, or a combination thereof; 3) Synthetic (i.e. Random Over Sampling Examples or ‘ROSE’, Synthetic Minority Oversampling Technique or ‘SMOTE’); and 4) cost sensitive or “error correct” learning. We intend to try all four methods to see which one performs best. There are quite a few articles on imbalancing [10 to 21]. In particular, [15] discusses undersampling, [15-18] the SMOTE technique and results, and [19-21] cost-sensitive methods. We will discuss these techniques in greater detail in the next section.

Finally, the review of literature that discusses data transformation and data scaling is now discussed.

Crone et al [22] do a high-level review of 16 research papers and detail various pre-processing steps done by the various authors, including the methods used (i.e. CART), data reduction (feature selection), standardization and/or discretization of the continuous attributes. Further, a standard method of scaling is discussed (z-score) which we will use in our data. Discretization, or ‘binning’, of categorical data is also mentioned and this is a method we will use on our data.

# Dataset

There are two datasets in the University of California at Irvine Machine Learning Repository [5]. The bank dataset was collected by S. Moro et al [1]. The end result is to obtain a model or models that will provide an effective telemarketing strategy to sell long-term deposit accounts. Data from old and current telemarketing campaigns were obtained as part of this dataset. Multiple contacts were often needed to determine whether a customer would subscribe to a long-term deposit account. The data is described below in Table 1. The steps used are best summarized in the diagram (Figure 1) below:

*Figure 1: Data Pre-Processing*

The ‘full’ dataset has 45,211 examples and is ordered by date from May 2008 to November 2010. The second bank dataset has only 10% of the examples (4,521) randomly selected from the full dataset to allow for testing more computationally demanding machine learning algorithms. We will be using the full dataset as much as possible. The dataset attributes are shown below in Table 1.

Table 1: Data List

|  |  |  |  |
| --- | --- | --- | --- |
| # | Attribute | Description | # of classes for categorical data |
| 1 | Age | Age | numeric |
| 2 | Job | Type of job | 12: ‘admin’, ‘blue-collar’, ‘entrepreneur’, ‘housemaid’, ‘management’, ‘retired’, ‘self-employed’, ‘services’, ‘student’, ‘technician’, ‘unemployed’, ‘unknown’ |
| 3 | Marital | Marital status | 3: married’, ‘divorced’, ‘single’ (divorced includes widowed) |
| 4 | Education | Level of Education | 4: ‘primary’, ‘secondary’, ‘tertiary’, ‘unknown’ |
| 5 | Default | Has credit in default or not | 2: yes, no |
| 6 | Balance | Average yearly balance in Euros | numeric |
| 7 | Housing | Has housing loan or not | 2: yes, no |
| 8 | Loan | Has personal loan or not | 2: yes, no |
|  |  | ***Related to current campaign:*** |  |
| 9 | Contact | Type of communication | 3: ‘cellular’, ‘telephone’, ‘unknown’ |
| 10 | Day | Last contact day of the month | numeric |
| 11 | Month | Last contact month of the year | 12: ‘jan’, ‘feb’, ‘mar’ …. ’nov’, ‘dec’ |
| 12 | Duration | last contact duration in seconds | numeric |
| 13 | Campaign | # of contacts for this customer | numeric |
|  |  | ***Related to previous campaign:*** |  |
| 14 | Pdays | # of days since previous campaign (-1 means client was not previously contacted) | numeric |
| 15 | Previous | # of contacts before this campaign | numeric |
| 16 | Poutcome | Outcome of the previous campaign | 4: ‘failure’, ‘success’, ‘other’, ‘unknown’ |
| 17 | Y | **Class attribute showing whether the client has subscribed a term deposit or not** | **2: yes, no** |

There is a total of 17 attributes with 7 numeric features and 10 nominal / categorical features (of which four are binary – Default, Housing, Loan and the “Y” target). The purpose of each feature is described in Table 1. The numeric features are Age, Balance, Day, Duration, Campaign, Pdays and Previous. The categorical features are Job, Marital, Education, Contact, Month, Poutcome, and the four binary features previously noted. The next two Tables further detail the attributes to get a better understanding of the data itself. There was no missing data, and no obvious incorrect data.

Table 1 above shows the categorical attributes and it is interesting to note that there is an “unknown” category in four of these attributes (job, education, contact, and poutcome). The ‘job’ category has 12 classes, which is a lot to discern from. The ‘married’ class has only three classes. Education has four, but one of them is ‘unknown’. Contact is either by ‘cellular’ or regular ‘telephone’, and also ‘unknown’. Month has the 12 months of the year. Poutcome has ‘failure’ and ‘success’ but also ‘other’ and ‘unknown’ again. And finally, the binary attributes are either ‘yes’ or ‘no’ for Default, Housing, and Loan.

Table 2 shows the numeric data with the minimum value, Q1, median, mean, Q3, the maximum value and the standard deviation. We will show plots, after the tables, that show that all the numeric data is not normal and is right-skewed. The outliers noted will **NOT** be taken out of the dataset. They were mostly on the upper end, and all will be normalized, when required, for the models that need it. Also,

Pdays contains a “-1” and even though it doesn’t make sense in the definition, and “0” doesn’t either, ***all values of “-1” will be replaced with zero*** (which currently doesn’t exist in this attribute) *knowing that 0 means NO CONTACT and not 0 days.* This attribute can then be normalized (0 to max) for models.

Also, when pdays = -1, previous = 0, which makes sense, but this means that the two variables are correlated when they shouldn’t be. We will see the correlation table further down and note if there are any large correlations. ***In ‘previous’, there is one exceptional last value of 275 that will be deleted****.*

Table 2: Numeric Attributes Information

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Attribute | # outliers | Min | Q1 | Median | Mean | Q3 | Max | Std Dev. |
| 1 | Age | 487 | 18 | 33 | 39 | 40.94 | 48 | 95 | 10.6 |
| 6 | Balance | 4729 | -8019 | 72 | 448 | 1362.3 | 1428 | 102127 | 3044 |
| 10 | Day |  | 1 | 8 | 16 | 15.8 | 21 | 31 | 8.3 |
| 12 | Duration | 3235 | 0 | 103 | 180 | 258.2 | 319 | 4918 | 257.5 |
| 13 | Campaign | 3064 | 1 | 1 | 2 | 2.8 | 3 | 63 | 3.1 |
| 14 | Pdays | 8256 | -1 | -1 | -1 | 40.2 | -1 | 871 | 100.1 |
| 15 | Previous | 8256 | 0 | 0 | 0 | 0.6 | 0 | 275 | 2.3 |

Finally, it is best to get a sense of which classes within the attributes dominate, if any. For that, Table 3 below shows the breakdown for any category over 20%. We see that there is definite dominance of categories in quite a few of the attributes. We further checked to see what, if any, dominance it would also have for the target answer of ‘yes’ and found that for ‘Default’ it was 99%.

Table 3: Attributes with Dominant Categories contributing over 20% of the outcome

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Attributes | Category | Dominant category | For same dominant category, what % is for “yes” answer only? |
| 1 | Age | 31-40  41-50 | 39%  25% | 34%  19% |
| 2 | Job | ‘blue-collar’  ‘management’ | 22%  21% | 13%  25% |
| 3 | Marital | ‘married’  ‘single’ | 60%  28% | 52%  36% |
| 4 | Education | ‘Secondary’  ‘tertiary’ | 51%  29% | 46%  38% |
| 5 | **Default** | ‘no’ | **98%** | **99%** |
| 6 | balance | -8,019 to 10,000 | 98% | 97% (range is too large; leaving as is) |
| 8 | Loan | ‘no’ | 84% | 91% |
| 9 | Contact | ‘cellular’  ‘unknown’ | 65%  29% | 83%  10% |
| 11 | Month | ‘may’ | 30% | 17% |
| 12 | Duration | 0 to 300 | 73% | 35% |
| 13 | Campaign | 1 to 10 contacts | 97% | 99% (range is too large) |
| 14 | Pdays | -1 -> 0 | 81.7% | 64% |
| 15 | Previous | 0 to 25 | 99.9% | 99.9% (range is too large) |
| 16 | Poutcome | ‘unknown’ | 82% | 64% |
| 17 | Y | **‘yes’** | 89% | n.a. |

Based on the above information, the attribute ‘Default’ will be deleted from the dataset as the category ‘no’ is extremely high at 98% of all instances. ‘Balance’, ‘Campaign’ and ‘Previous’ will not be removed as we realized the range we used was too large. This Table was created using Excel (only one) and so doing this was laborious. We note that the ‘target’ is highly imbalanced, with the “yes” answer only 12% of the responses.

The visualization of the data is next. All tests and plots were performed in R code.

Numeric data

In order to perform numerical analysis, the data is presumed “normal”. This implies that the data has a bell-shaped curve (not skewed), and does not have multicollinearity (the dependent variables are not correlated). In reviewing the data in R, we did normality tests, boxplots, histograms, and a correlation map for the numeric data. Q-Q plots were also run in R (not showed here) and they distinctly show that the data is not normal.

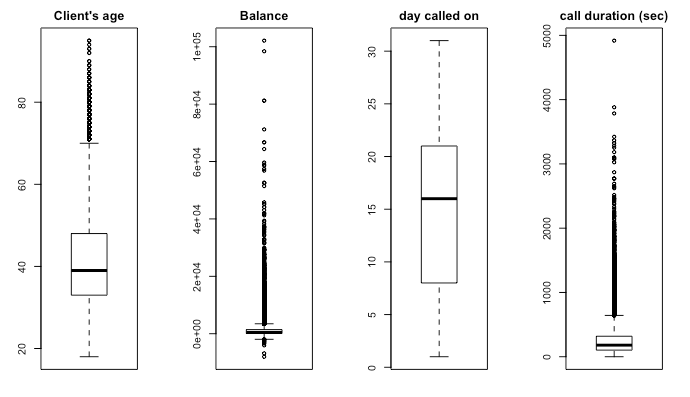
As well, the normality of each numeric attribute was tested using the Shapiro-Wilk test (not shown), and all tests failed, meaning that the numeric attributes were not normal (any value above 0.05 indicates normality). Therefore, the data will be normalized or scaled only for the models that require it. The histograms also show the skewness of all the numeric attributes.

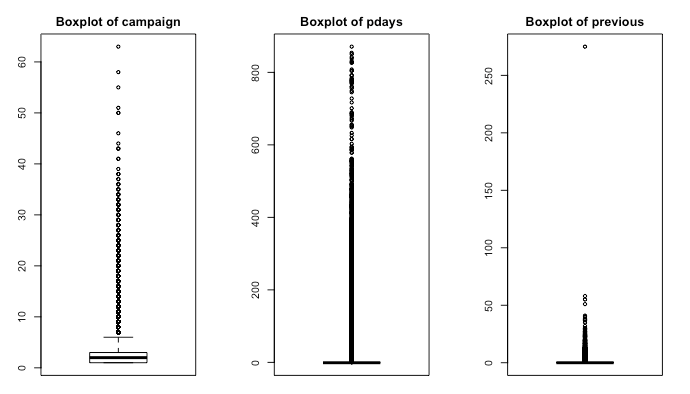
Categorical data

As well, correlations were performed for the categorical data using Pearson’s Chi-squared test and the tests showed that the data was relatively unrelated to each other. The work was done in R and is not shown here.

The boxplots of the numeric data are shown below in Figure 2. From these plots, we can see that the data is not normal, based on the size and location of the boxes. One can see the outliers such as in ‘previous’.

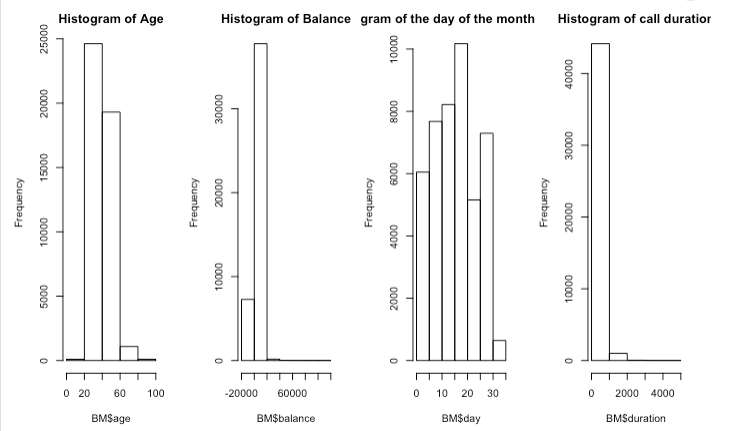
*Figure 2: Boxplots of Numeric data*

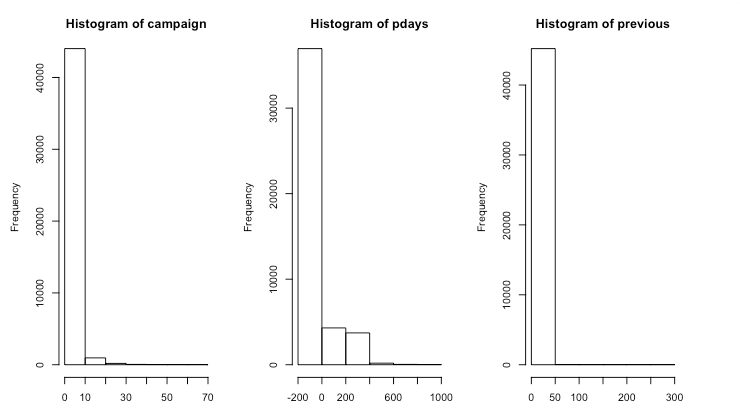




The histograms of the numeric data are shown below in Figure 3. It is obvious from the histograms that all the numeric data is not normal and all were skewed to the right, however this can often happen with real data. It is however required for some models, such as regression, that the data is normal. Hence, the numeric data will be normalized or scaled, where required, for the models that need that (i.e. k-NN or k-means). Notice that for many of the attributes, a large portion of the instances are on the far left.

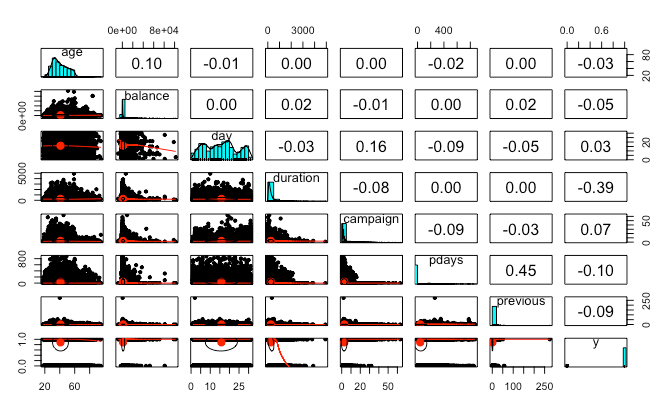
*Figure 3: Histograms of Numeric data*





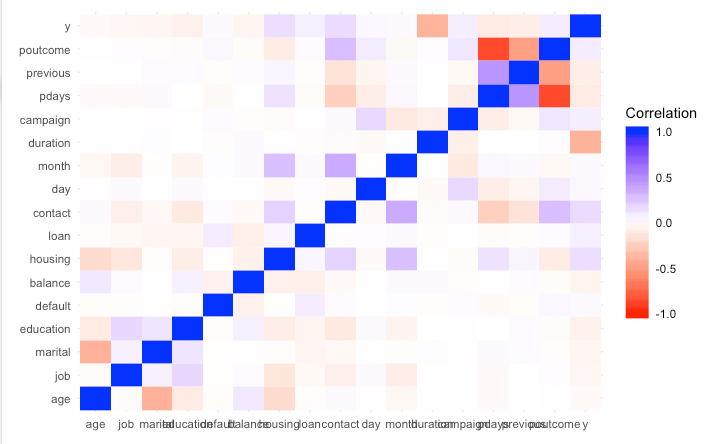
We have a screenshot of a plot from R of the numeric data below in Figure 4. The upper right area shows the correlations, the diagonal shows the histograms, and the bottom left area shows the scatterplots with the circles showing the strength of correlation (circle means a little coorelation, and the oval means a lot). Correlations were performed for the data to ensure there was no multicollinearity – but also to see if there was any correlation between the data and the result, ‘Y’. By reviewing the correlations, we see that the data has low correlations with each other (all under 0.16, one at -0.39) except for pdays to previous at 0.45, which is still an acceptable correlation. Note that the data does not have strong correlations to the “y” answer!

*Figure 4: Correlations/Histograms/Scatterplots of Numeric data*



A correlation heat map of all the data was also generated for a visual representation of the correlations (see Figure 5 below). Upon review, the data correlations showed that they were all very low, except for three stronger negative correlations – in R code (Y to duration is -0.39, poutcome to previous of -0.49, poutcome to pdays of -0.86).

*Figure 5: Correlation Heat Map of all data*



To summarize, the value of ‘-1’ has been replaced with ‘0’ in pdays, the value of ‘275’ has been deleted from ‘previous’, and the attribute (#5) ‘default’ has been deleted. The dataset has now been adjusted and will be used ‘as is’ where possible in the models. However, if any conversions need to be made, they are discussed in Step 1 of the next section. The R source code is stored in Github for the Dataset section [see link – <https://github.com/njeanius/bank1/tree/draft> ].

# Approach

The block diagram is shown below. We used a modified version of the CRISP [2,3] methodology. Because the data steps were covered above under Dataset section, they were taken out of this diagram.

The data is a binary classification problem in that (some of) the inputs (both numeric and nominal) should be able to predict the outcome of the class variable (Yes or No) based on the respective models. The steps are broken down as follows.

*Figure 6: Diagram of steps in Modeling Data*

COMPLETED TO HERE….the rest is in draft form, and may be incorrect at this time…

## Step 1: Pre-Processing

Note: not sure yet of the order in which I want to do this as it creates too many datasets….

## Step 1A: Rebalance the data but also keep Unbalanced set

Because the data was imbalanced, with a 89% “no” and only 11% “yes” in the target, rebalancing should be done. The various methods to do this include:

1. Undersampling the majority class
   1. Random
   2. Informative (EasyEnsemble and BalanceCascade)
2. Oversampling the minority class
3. Under- and over-sampling
4. Synthetic Data Generation = ROSE, SMOTE, SMwR? with bootstrapping and k-nearest neighbors
5. Cost Sensitive Learning- choose a classifier with lowest total cost where Total Cost = C(FN)xFN + C(FP)xFP

ROSE (Random Over Sampling Examples) package helps us to generate artificial data based on sampling methods and smoothed bootstrap approach. This package has well defined accuracy functions to do the tasks quickly.

WEKA:

1. the ClassBalancer function, (best in Weka -what is it in R?)
2. SMOTE or Synthetic Minority Oversampling Technique
3. Cost sensitive or “error correct”

Under-sampling reduces the majority class. The sampling can be done 1) randomly or 2) informatively. For the second method, there is EasyEnsemble or Balance Cascade. EXPLAIN.

Over-sampling increases the minority class. The sampling can be done 1) randomly or 2) informatively. With the second method, overfitting can result along with reduced accuracy.

A combination of the two can also be done just by changing a parameter in the R code.

For synthetic data generation, a well-known technique is SMOTE (synthetic minority oversampling technique). Other methods are bootstrapping combined with k-NN, ROSE>>?? We will use SMOTE, ROSE…

For cost-sensitive learning, we don’t create any balance data, but basically “punish” the model using a cost or confusion matrix. We usually punish for getting a FP (?) and this is the worst case because we wouldn’t want to give a client any credit if they end up bankrupt. On the other hand, if we miss giving a client, who would use the money wisely, is not as big a deal. Discuss the process.

We also have to think about the training and test set. We could do a 70/30 split using sampling technique, and verifying the sample is the same percentage for UNBALANCED DATA as the original dataset.

If the sample size is small, 10-fold cross-validation can be used.

However, we will also take a sample of the original data and use that for validation in the evaluation stage. We will ensure that none of the data is replicated in the test, training or validation sets.

## Step 1B: Scaling Numeric data and Normalizing data

There are four methods that can be used to scale numeric data. They are called: 1) the standardized method using z-scores; 2) the mean normalize method; 3) min-max scaling; 4) and unit vector. Methods x and 3 are quite common and we will use these methods, (3: zero to 1). Scaling is good when data has ranges that are very different from one another. Scaling is not normalizing data. Normalizing data changes the shape of the data to a bell curve whereas scaling just scales down the numbers so that they are between 0 and 1 (in this case). This version will only be used when the models require a scaled distance. This dataset is called PB\_scale. K-means, K-NN, and PCA require scaling.

min-max scaling is:

z-scalar: data is -1 to 1

formula is: z-score = x' = (x- mean)/st.dev.

USE BOTH

As for normalizing, the R command “normalize(dataset, lambda=3)” can be used. This dataset will be used where required. This dataset is called PB\_norm.

## Step 1C: Convert Numeric to Nominal (discretize) and vice versa for a new set of data

Notes here: can we use r code to convert: Numeric to categories: Discretize() ??

Numeric to Nominal - We have added R code to convert the original numeric data to nominal data. Table 5 shows the bins that are used to convert the number of data items into separate bins. For example, the first bin in Age is from 0 to 30 years old, so the number of data items in that group will go into the first new nominal attribute. ---\*\*\*\* there is an alternative that uses quartiles \*\*\*\*.

Table 5: When converting numerical data to Categorical data for models:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **# categories created using binning** | **comment** |
| Age | 6 | [0, 30, 40, 50, 60, 70, 100] |
| Balance | 6 | [-10000, 10000, 30000, 50000, 70000, 90000, 110000] |
| Day | 4 | [1, 8, 15, 22, 31] |
| Duration | 5 | [0, 300, 600, 900, 1200, 5100] |
| Campaign | 5 | [1, 5, 10, 20, 30, 70] |
| Pdays | 6 | [0, 1, 100, 200, 300, 400, 900] |
| previous | 4 | [0, 2, 5, 25,300] |

Nominal to Numeric - We have added R code to convert the original nominal data to numeric data. Table 6 shows the number of new attributes that will be created (n-1) from the number of categories in each original attribute. For example, Job has 12 original categories so 11 new “dummy” (0 or 1) attributes will be created from these.

\*\*\*\* or use “BM$marital<- as.numeric(BM$marital) - in R code for correlation matrix

Table 6: When converting Categorical data to numerical data for models:

|  |  |  |
| --- | --- | --- |
| Attribute | Current # categories | # Attributes created using ‘dummy’ variables (0,1) |
| Job | 12 | 11 |
| Marital | 3 | 2 |
| Education | 4 | 3 |
| Default | 2 | 1 |
| Housing | 2 | 1 |
| Loan | 2 | 1 |
| Contact | 3 | 2 |
| Month | 12 | 11 |
| poutcome | 3 | 2 |

## Step 2: Feature Selection or Dimensionality Reduction (Select Attributes)

There are various methods we can use to reduce the number of attributes (or features) in a dataset [ref: CMTH642 class notes week 9]:

1. Remove data with a lot of missing values (which this dataset doesn’t have).
2. Use a low variance filter – i.e. remove unused attributes like “gender” if all were, say, female (our dataset doesn’t have this).
3. Reduce highly correlated items (we don’t have although we didn’t check for correlation between numeric and nominal).
4. Run PCA on all numeric data and see the results.
5. Use Random Forest and see the results to reduce.
6. Use a Feature Selection algorithm.

We will look at methods 4, 5 and 6 to see if we can reduce features in the dataset before moving onto models.

Still use train and test set….

RFE / Recursive **Feature** Elimination in Caret – I did this already in the correlations test on numeric data

Boruta is a feature selection algorithm for finding all relevant variables. The algorithm is designed as a wrapper around a Random Forest classification algorithm.

GainRatioAttributeEval(formula, data, subset, na.action, control = NULL)

InfoGainAttributeEval(formula, data, subset, na.action, control = NULL)

## There are various Feature Selection models that can be used before running any models. These models include:

* Boruta, Lasso regression, Stepwise Forward and backward selection, RFE, Genetic

In WEKA:

**ChiSquaredAttributeEval, CfsSubsetEval**

Algorithms:

Statistical techniques on numeric data include Chi-Square score and Gini index. Meanwhile, most algorithms in this family can only work on discrete data and conventional data discretization techniques are required to preprocess numerical and continuous variables.

Others include Multilayer perceptrons (MLP), Greedy FS, Lasso

## Step 3: Modeling

The models we want to use are the classification models, and with supervised training. The supervised training means that we know the ‘target’ response and we want to train the data to that target so as to obtain an algorithm that works well.

The models used are: a) Naïve Bayes, b) Decision Tree, c) xx, d) xxx…..

Then describe each model

Bayesian Network (BN), Naïve Bayes (NB), NaiveBayesUpdateable (NBU), MLP, Simple Logistic (SL), SMO, Decision Table (DT), OneR, J rip, Decision Stump (DS), J48, Random Forest (RF), RandomTree (RT), REPtree (RepT).

Extra- time permitting-We will also use Unsupervised training, where the target has been stripped out of the data, to get an idea of any relationships in the data (describe these types – Apriori and clustering).

## 3A: Model 1 –

## 3B: Model 2 –

## 3C: Model 3 –

## 3D: Model 4 –

## 3E: Model 5 –

## 3F: Model 6 –

## Step 4: Model Evaluation

## Write details of the steps. If there is any source code that you’d like to share then provide the link of the Github.

## Step 5: Conclusion

## References

1. Moro, S., Laureano, R., and Cortez, P. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. P. Novais et al (Eds.), Proceedings of the European Simulation and Modelling Conference – ESM’2011, pp. 117-121, Guimaraes, Portugal, October, 2011.
2. Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. Decision Support Systems, 62, 22-31.
3. Moro, S., Cortez, P., & Rita, P. (2015). Using customer lifetime value and neural networks to improve the prediction of bank deposit subscription in telemarketing campaigns. Neural Computing and Applications, 26, 131–139.
4. Moro, S., Cortez, P., & Rita, P. (2018). A divide and conquer strategy using feature relevance and expert knowledge for enhancing a data mining approach to bank telemarketing. Expert Systems, 35(3).
5. Elsalamony, H. Bank Direct Marketing Analysis of Data Mining Techniques. International Journal of Computer Applications, V85, 12-22, 2014.
6. M. Purnachary, B. Kumar and H. Shaziya. Performance Analysis of Bayes Classification Algorithms in WEKA Tool using Bank Marketing Dataset. International Journal of Engineering Research in Computer Science and Engineering (IJERCSE), V5, Issue 2, February 2018.
7. P. Ładyzynski, K. Zbikowski and P. Gawrysiak. Direct Marketing Campaigns in Retail Banking with The Use of Deep Learning and Random Forests. Expert Systems with Applications, V134, 28-35, 2019.
8. J. Li, K. Cheng, S. Wang et al. Feature Selection: A Data Perspective. ACM Computing Surveys, Vol. 50, No.6 Article 94. Dec 2017.
9. M. Alibeigi, S. Hashemi, A. Hamzeh. Unsupervied Feature Selection Based on the Distribution of Features Attrbiuted to Imbalanced Data Sets. International Journal of Artificial Intelligence and Expert systems (IJAE), vol.2, issue 1. 2011.
10. S. del Rio, V. Lopez, J. Manuel et al. On the use of MapReduce for imbalanced big data using Random Forest. Information Sciences, 11/2014, Volume 285
11. S. Cateni, V. Colla, M. Vannucci. A method for resampling imbalanced datasets in binary classification tasks for real-world problems. Neurocomputing, 07/2014, Volume 135
12. C. Beyan, R. Fisher. Classifying imbalanced data sets using similarity based hierarchical decomposition. Pattern Recognition, 48 (5) (2015), pp. 1653-1672
13. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications, 73(Complete), 220–239.
14. V. Lopez, A. Fernandez, S. Garcia, V. Palade, F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics.Information Sciences, 250 (2013), pp. 113-141
15. Miguéis, V.L., Camanho, A.S. & Borges, J. Predicting direct marketing response in banking: comparison of class imbalance methods. Serv Bus 11, 831–849 (2017).
16. Sáez J.A., Krawczyk B., Woźniak M. Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets.Pattern Recognition, 57 (2016), pp. 164-178
17. Marinakos, G., & Daskalaki, S. (2017). Imbalanced customer classification for bank direct marketing. Journal of Marketing Analytics, 5(1), 14-30.
18. Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4), 221–232.
19. Jena, P. C., Ojha, A. C., & Pani, S. K. (2015). Cost-sensitive modeling to predict bank telemarketing campaign. IUP Journal of Information Technology, 11(2), 39-47.
20. Ghatasheh, N., Faris, H., AlTaharwa, I., Harb, Y., & Harb, A. (2020). Business analytics in telemarketing: Cost-sensitive analysis of bank campaigns using artificial neural networks. Applied Sciences, 10(7), 2581.
21. F. Cheng, J. Zhang, C. Wen, Z. Liu, Z. Large cost-sensitive margin distribution machine for imbalanced data classification.Neurocomputing, 224 (2017), pp. 45-57
22. S. F. Crone, S. Lessmann, and R. Stahlbock. The impact of preprocessing on data mining: an evaluation of classifier sensitivity in direct marketing. In European Journal of Operational Research, vol. 173, no. 3, pp. 781–800, 2006.
23. <http://archive.ics.uci.edu/ml> (The UCI Machine Learning Repository collection of datasets).