ISDS-574 - Data Mining for Business Applications

To

Dr. Yinfei Kong

Predicting: The Success of Bank Telemarketing by Data-Mining Approaches

From (Group 3):

Anh Phan
Anand Panchal
Apurva Desai
Ankita Upadhyay
Kedar Kulkarni
Vaishali Patel
Tongying Shen

Contents

1. Executive Summary

The Portuguese bank has successfully operated a telemarketing campaign where all records were collected into a data set. The purpose of our analysis is to classify the potential customers from the telemarketing campaign who would subscribe to long-term deposits. Data Mining methods used in this paper are taught in ISDS-574 course and influenced by research authors, which are mentioned in Literature Reviews.

The paper opens with an introduction about the Portuguese banking system and the purpose of a telemarketing campaign. Next, it explains the data set and question of interest in detail. Since the dataset that we obtained are very large, data preprocessing and exploration steps are critical to our analysis purpose. The following steps were performed: Reducing Categorical Variables, Creating Dummies, Removing References, Finding Correlation and Detecting Outliers. The final data set contains relevant variables, has no missing values and little collinearity, and is ready for further analysis.

In this paper, our team implemented three classification techniques: k-Nearest Neighbors, Decision Trees and Logistics Regression. The results from these methods are compared using misclassification error rate and sensitivity rate. The highlight of our project has been using a cutoff of 25 percentages instead of the majority voting of 50 percentages. A lower cutoff is believed to produce higher sensitivity, which effectively improves the chance of identifying the success class for telemarketing purpose.

We conclude that logistics regression offers the best model to classify our future customers, based on the highest sensitivity and a decent misclassification error rate. However, the best technique should be chosen based on task priority, project's budget and time efficiency. Data Mining applications in real life should not only be driven by numbers, but also by the business's needs and resources. Managers should have insights about their business to determine which classification method to use.

2. Introduction

2.1 Background

The Portuguese banking system has witnessed significant structural changes over the last three decades, with a shift from a government-controlled system to a market-driven environment fully integrated with the European Union. The Portuguese economy is relatively small and highly integrated into the Eurozone, therefore susceptible to adverse market conditions. The Portuguese banking sector, like other sectors of the Portuguese economy, has been affected by the Eurozone debt crisis. The recent crisis in international financial markets and the resultant global economic slowdown has led to unusually unfavorable conditions for banking activities worldwide.

Due to the economic downturn and competition in the banking system, a Portuguese bank decided to operate a marketing campaign to sell long-term deposits. Long-term deposit accounts sold by the bank are where customers can securely invest their money. Such deposits require a fixed investment that includes a guaranteed interest rate for the agreed period. This means if the market crashes, customers' money are safe, unlike the share market. The bank

promoted these accounts through a direct marketing campaign at its call center, where agents make call to clients or client's call in to make inquiries. Throughout the campaign, data were collected from 2008 to 2013, thus including the effects of the global financial crisis in 2008.

In time of crisis, marketing is one of the most important functions to enhance a business. To effectively target customers, most companies use telemarketing. Such marketing is usually performed by a call center, where all remote interaction with customers are centralized and tracked. Contacts at a call center are divided into "Outbound" or "Inbound". When an agent initiates the conversations to sell a product or service, it's called "Outbound marketing". When a client calls in the center for any reason and is asked to buy a product, it's called "Inbound marketing". Both methods constitute large marketing budgets for many businesses, including payroll expenses, facility costs, etc. Therefore, reducing costs using technology is critical. With the development of technology, the efficiency of telemarketing is enhanced significantly. For example, through the evaluation of available information and customer preferences, a company could focus on maximizing customer values and building strong relationships that matter to its business goals. Specifically, with limited resources on financial and human capital, a company's marketing campaign could select the best set of clients that are more likely to subscribe to an offer, thus increase success rate and reduce cost.

In this paper, we analyze the telemarketing data from the Portuguese bank and propose a few models that can classify a customer who would likely open a long-term deposit account. These models are valuable to help bank managers prioritize and select the next customers to be contacted during the campaign. As a result, time and expenses for the campaign could be greatly reduced. Additionally, by contacting customers who are likely interested, bank agents could effectively sell term deposits to customers and customers' stress and intrusiveness could be diminished.

2.2 Data Description

The dataset we use in this study was obtained from UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems. There are 45,211 records in the original data set, including the client information (8 variables), the last contact information (4 variables), other attributes from the campaign (4 variables) and the campaign results, ordered from May 2008 to November 2010. No missing value included. The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y). Please refer **Figure 1** in **Appendix I** to see the first 20 records of the original dataset.

2.3 Question of Interest

This project classifies the potential customers from a telemarketing campaign who would subscribe to long-term deposits. We would like to classify whether the customer will accept the long deposit scheme or not based on Age, Education, Marital Status, Job, and so forth. Another aspect of this analysis is to interpret the data-driven model to find out what are significant

predictors of this outcome and understand which factors best identify a customer who would accept the long-term deposit and which factors do not help in this classification. Using the classification models, the Portuguese bank could deploy its limited resources to most valuable customers and generate the best success rate from the campaign.

2.4. Literature Reviews

1. Upon getting the dataset from UCI Machine Learning website, we wanted to understand how the data was collected. S. Moro, P. Cortez and P. Rita (2014) examined a large dataset (52,944 phone contacts) from a Portuguese retail bank. The authors propose a personal and intelligent Decision Support Systems (DSS) that uses a data mining approach for the selection of bank telemarketing clients. The records were divided by time order, which produced a training set of 4 years and a test data of one year. The authors merged an external dataset that was rich in social and economic influence features obtained from central bank of the Portuguese Republic Statistical web site. The merged data set had a total of 150 variables. However, after the feature selection procedure, they collected a reduced dataset of 22 relevant attributes. Also, 4 methods were used: Logistics Regression, Decision Trees, Neural Networks, and Support Vector Machines. These methods were compared using 2 metrics, area of the receiver operating characteristic curve (AUC) and area of the LIFT cumulative curve (ALIFT). In conclusion, Neural Networks produced the best result with an AUC of 0.80 and ALIFT of 0.67.
2. Vajiramedhin and A. Suebsing (2014) examined the Portuguese bank dataset from UCI Machine Learning Repository which is the same dataset that we are using in our case study. They have proposed that in order to get a better true positive rate % and ROC rate % for the case study one should use correlation-based feature subset selection algorithm and dataset balancing technique. They compared their proposed solution results to method 1 and method 2. Method 1 employed using all the features without using dataset balancing technique and feature selection algorithm. Method 2 employed using the correlation-based feature selection algorithm without the dataset balancing technique. The proposed solution yielded the best TP rate % and ROC rate % with 92.14 % and 95.60 % respectively. We have employed a similar approach in our case as well. We have used a balanced dataset technique and feature selection algorithms.
3. S. Moro, R. Laureano, and P. Cortez (2011) examined the Portuguese bank marketing campaign data using the CRoss-Industry Standard Process for Data Mining (CRISP-DM) methodology for increasing the success of data mining projects. The methodology defines six phases that includes Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment. These six steps help build and implement a data mining model in the industry and support critical business decision. They employed Naive Bayes, Decision Tree, and Support Vector Machine algorithms and evaluated these algorithms by using Area Under the Curve (AUC) and Area Under the LIFT Curve (ALIFT). The SVM method provided the best AUC and ALIFT results with values of 0.938 and 0.887 respectively. To achieve our goal, we used our understanding of the telemarketing business and tried to employ different cutoffs to get better results.

3. Data Pre-processing and Exploration

The Telemarketing dataset originally contained 42,511 records with 17 variables. With a dataset this large, the data exploration and pre-processing step is critical. The importance of data cleaning in the mining process cannot be undermined. It also might contain outliers. Before using the different classification models, we check if the dataset contains quality data and not just quantity, we perform extensive data cleaning steps on the original dataset. In the original dataset, "yes" outcome is 11.69% (5289 out of 45,211 records) of the total records, which is unfavorable in data modeling. Hence, we have created a subset of the dataset where an equal number of the outcome (Yes and No) records are selected randomly. Please refer R code used for creating the subset.

3.1. Reducing Categorical Variable

The dataset does not have missing values; hence, we will directly jump to the categorical variables. The categorical variables have different categories, and all the categories should be considered as an individual variable in the modeling of the models so that we would not miss any data during the modeling. In the dataset: *job*, *marital*, *education*, *contact*, *month* and *poutcome* are the categorical variables. The job and month variable have 12 different categories. To simplify the modeling, we have created four categories of job and month; the variables are reduced by level and the levels are indicated by the numbers-1,2,3 and 4. **Figure 1** summaries the categorical variable reduction.

Figure-1: Categorical Variable Reduction

Month		Job	
Q1	"jan", "feb", "mar"	White Collar	"admin", "management", "self-employed", "technician", "services.", "entrepreneur"
Q2	"apr", "may", "jun"	Blue Collar	"housemaid", "blue-collar"
Q3	"jul", "aug", "sep"	Other	"unemployed", "student", "retired"
Q4	"oct", "nov", "dec"	Unknown	"unknow"

3.2. Creating Dummy Variables

A Dummy variable or Indicator Variable is an artificial variable created to represent an attribute with two or more distinct categories/levels. In our data exploration, we have created dummy variables for all the categorical and binary variables. Binary variables are those variables which take only two values (e.g., yes and no). The table shown below gives details about the dummy variables created from the dataset variables.

Figure-2-: Binary and Categorical Variables

Binary variables		
Variable	Dummy variables	
default	default_no	default_yes
housing	housing_no	housing_yes
loan	loan_no	loan_yes
y (Output variable)	y_no	y_yes

Categorical Variable				
Variable	**Dummy Variables**			
marital	marital_divorced	marital_married	marital_single	
contact	contact_cellular	contact_telephone	contact_unknown	
job	Blue Collar	White Collar	Other	Unknown
education	education_primary	education_secondary	education_tertiary	education_unknown
month	Q1	Q2	Q3	Q4
poutcome	poutcome_failure	poutcome_other	poutcome_success	poutcome_unknown

3.3. Removing Reference

The number of dummy variables necessary to represent a single attribute variable is equal to the number of levels (categories) in that variable minus one. For categorical variables, we have determined the reference by considering the frequency of each dummy variable, and the highest frequency dummy variable is considered as the reference of the variable. For binary variables, the dummy variables with no values are considered as a reference. The frequency of each categorical variables and their dummy variables are tabulated in the table. The frequency table is generated using pivot table feature in Microsoft Excel. Also, the removed variables called as reference variables are listed in the table.

Figure-3: Removing Reference

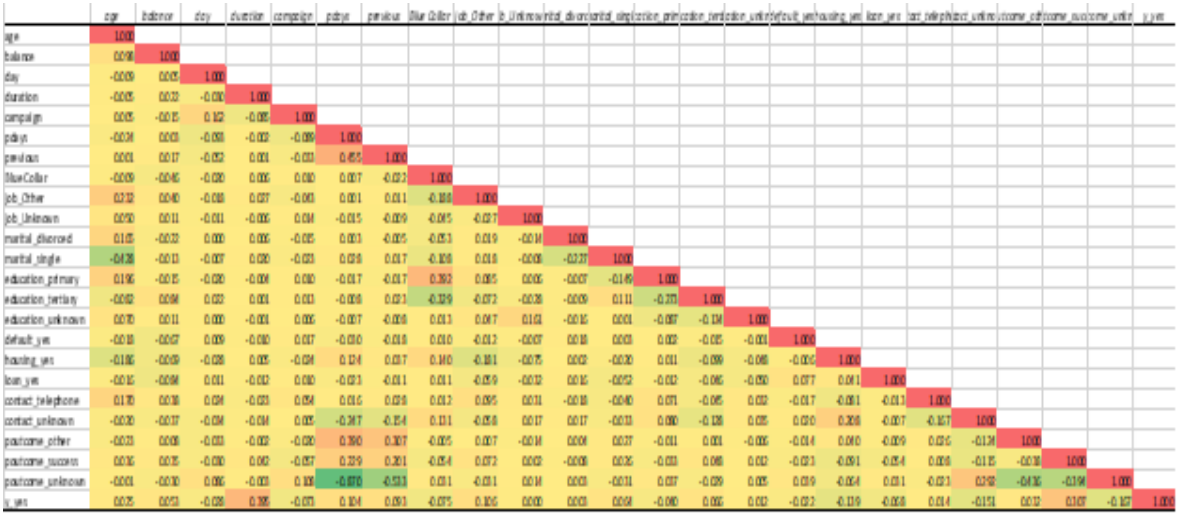
Variable	Dummy variables	Frequency
Job	White Collar	6914
	Blue Collar	2157
	Other	1445
	Unknown	62
Marital	Single	3370
	Married	5951
	Divorced	1257
Education	Primary	1400
	Secondary	5274
	Tertiary	3449
	Unknown	455
Contact	Cellular	7623
	Telephone	752
	Unknown	2203
Month	Q1	1328
	Q2	4777
	Q3	3186
	Q4	1287
poutcome	Success	1059
	Failure	1194
	Other	498
	Unknown	7827

Variables	Reference variables
job	White Collar
marital	marital_married
education	education_secondary
contact	contact_cellular
poutcome	poutcome_failure
month	Q2
housing	housing_no
loan	loan_no
default	default_no
y	y_no

3.4. Finding Correlation

After removing reducing categorical variables, we decided to check whether there exists multicollinearity. Multicollinearity refers to the correlation between variables. To check the multicollinearity, we developed a correlation matrix on the variables. Please refer Output for the output of Correlation Matrix (Cut off = 0.5) below:

Figure-4: Correlation

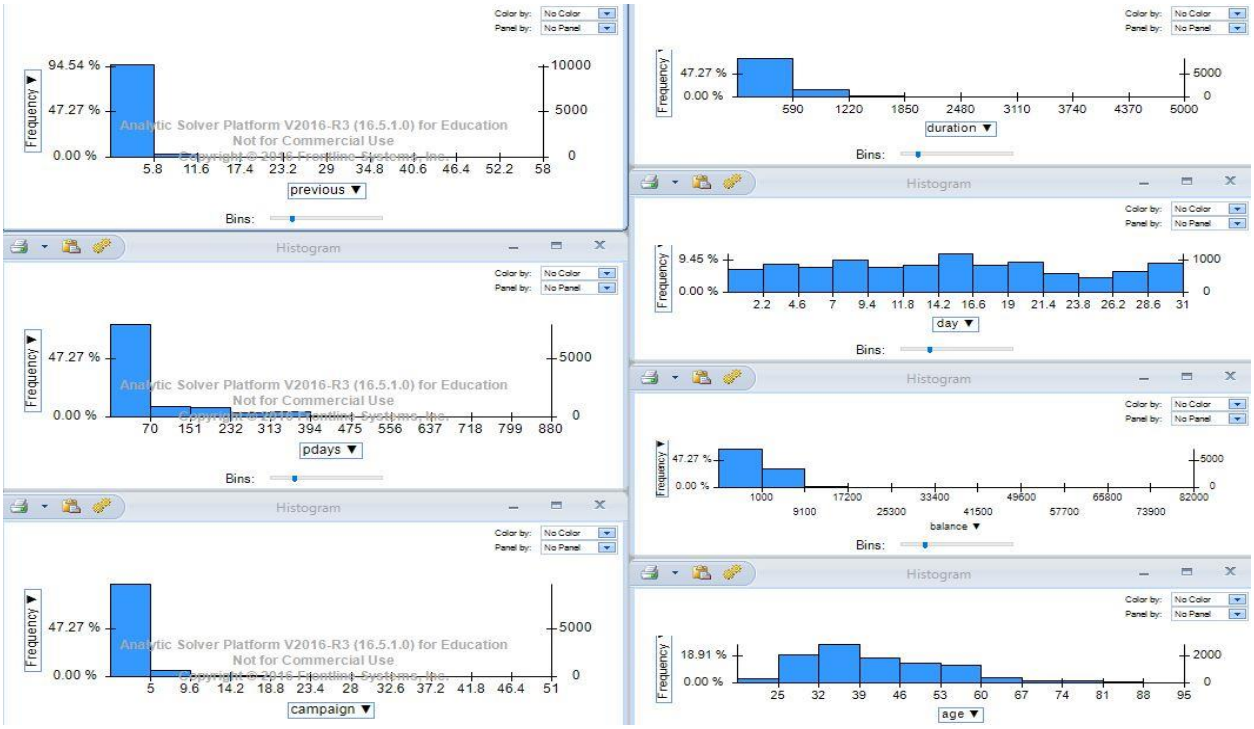


The second half of the output highlights the correlation of variables with themselves. If we carefully look at the correlation matrix output in the first screenshot, it highlights few cells. When we looked at these highlighted cells, we noticed that there exists obvious correlation between two variables. The variables are poutcome_unknown and pdays with a correlation of -.870. By the description of the variables, they are not harmful to keep in the dataset. Hence, we have not removed one of the variables for the analysis.

3.5. Detection of Outliers

To identify any outlier with variables, we have used visualization techniques for each numerical variable. The histogram shows the distribution of the variable against the frequency. We are looking for normal distribution of the variables. To check whether there exist any outliers in the variables, we have also plotted boxplot in the Visualization below. We learned from this distribution that there are no outliers in our dataset for the variables. All the histograms and boxplots are placed below:

Figure-5: outliers



4. Methods

Since the goal of this project is to identify customers who are likely to subscribe to a long-term deposit, it is critical to correctly identify the success case of "1" or "yes". It means that high sensitivity rate is the most important indicator of the best classification model. For this goal, we are willing to tolerate higher error rates for all methods as long as it is reasonable. We ran all methods with majority voting first and compared the results with 25% cutoff results. We only noticed small differences in error rates, but significant increases in sensitivity. Therefore, lowering our cutoff probability in each model would produce better results for our marketing purpose. In our telemarketing project, this means that even if there is only 25 percent chance of a customer to accept the offer, we would still contact the person. The cut off with 25 percentage will be applied to all following methods.

4.1 k-Nearest-Neighbors (kNN)

kNN is a very simple, yet powerful classification algorithms that works well in practice. For our study, we run kNN for the data on XLMiner with with normalized values and a cutoff of 0.25. No variables selection was performed because we found little collinearity among these variables and most of our variables are categorical that are transformed to binary. We also indicated that k max is 20 on XLMiner. Best k was chosen based on the lowest error rate. Ideally, best k is an odd number if there are two binary classifications.

4.2 Classification Trees (CART)

Classification is a data-driven method we have used for classification called Classification Tree. (Shmueli 186). We have decided to use the best prune tree for our paper. The 'best pruned tree' is the "smallest tree in the pruning sequence that has an error within one standard error of the minimum error tree" (Shmueli 204). Also, pruning is a useful strategy for avoiding overfitting of data. We have used the clean data with 26 variables (refer data summary table in data processing) to create a best prune tree. We derived the Best Prune at cutoff probability value for success 0.25, which gave us 6 nodes (refer best prune tree in results). In our case, the best prune tree is obtained by adding one standard error (0.61%) to minimum to the validation error of minimum error tree (19.7%). The smallest tree with the validation error below the total of 19.8% is the best prune tree.

There is no variable selection in CART. But it performs recursive partitioning that split the data into non-overlapping multidimensional rectangles. This method is computationally cheap to deploy even on large samples. They also have other advantages such as being highly automated, robust to outliers, and can handle missing values.

4.3 Logistic Regression (LR)

Logistic regression is a powerful model-based tool. Under the logistic regression method, here we apply three different variable selection methods - forward selection, backward elimination, and stepwise selection to help us find the best model possible. First we run the models using XLMiner. It recommends the empty model under forward selection and stepwise selection, and it also shows that the best model under backward elimination is the full model; the results are not useful for our purpose of study, since we are trying to build a model that has some variables but have less variables than the full model. Then, we use R to perform the logistic regression, and R gives more reasonable models here. For forward selection method, it shows a model with 18 variables out of the 26 variables; for backward elimination method and stepwise selection method, R shows the same model which has 17 variables out of the 26 variables, and the only difference is the order of the variable selected. However, XLMiner and R give the similar error rate, sensitivity and specificity.

The reason why the two methods have different results in terms of regression models is that the two-software's use different standards for variable selection. In XLMiner, the selection is based on the F value. Under forward selection, for a variable to come into the regression, the statistic's value must be greater than the value of F-to-enter (FIN, default = 3.84). Under backward elimination, for a variable to leave the regression, the statistic's value must be less than the value of F-to-remove (FOUT, default = 2.71). Under stepwise selection, for a variable to come into the regression, the statistic's value must be greater than the value for FIN (default = 3.84); for a variable to leave the regression, the statistic value must be less than the value of FOUT (default = 2.71); and the value for FIN must be greater than the value for FOUT. On the other hand, R picks the best model using Akaike information criterion (AIC) score, the smaller the AIC, the better quality of the model.

5. Results and Interpretation

5.1 k-Nearest-Neighbors (kNN)

According to XLMiner result, the best k is 5 and the error rate on validation data is 23.86578%. The model produces a high sensitivity rate of 88.24%, which is suitable for our study purpose. The area under ROC Curve is considered good, as it has a large margin with approximately 84%. The k in kNN tells the algorithm how many nearest neighbors, in terms of Euclidean distance in feature space, should be used to determine the class of an unknown data point, in this case by 25% voting. In this case, let's say we have a new customer that needs to be classified to be contacted. We look for the 5 nearest neighbors and find out that 2 out of 5 (40%) customers did subscribe to the long-term deposits. This new customer would be labeled as "likely to subscribe" as kNN model forms more than 25% voting (40%>25%). This helps bank managers decide to contact this customer because the chance of success is high.

Figure-6-:k-NN output

Value of k	% Error Training	% Error Validation	
1	0	25.1654064	
2	12.2124173	25.9215501	
3	18.2634731	27.5283554	
4	23.5896628	29.6550095	
5	17.3495115	23.8657845	<- Best k = 5
6	20.2332178	25.3544423	
7	22.4550898	26.2996219	
8	24.4248345	27.5519849	
9	20.895052	24.4801512	
10	22.4866057	25.6852552	
11	23.8417901	27.1502836	
12	25.1497006	28.1899811	
13	22.4866057	25.8742911	
14	23.7630003	26.5831758	
15	24.9133312	27.362949	
16	25.9060826	28.1190926	
17	24.0624015	25.9215501	
18	25.1497006	26.9612476	
19	25.764261	27.7173913	
20	26.6309486	28.426276	

Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	1884	251
0	759	1338

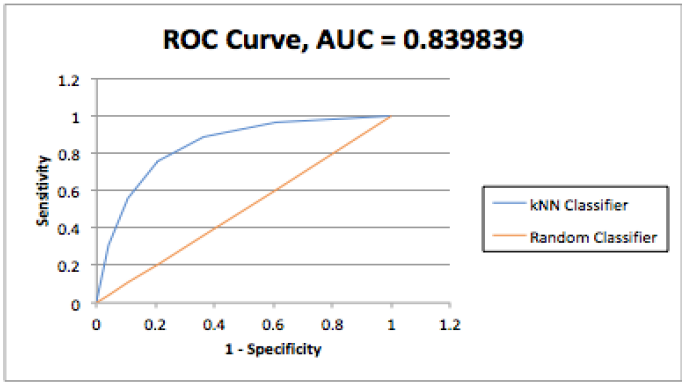
Error Report

Class	# Cases	# Errors	% Error
1	2135	251	11.7564403
0	2097	759	36.1945637
Overall	4232	1010	23.8657845

Performance

Success Class	1
Precision	0.71282633
Recall (Sensitivity)	0.8824356
Specificity	0.63805436
F1-Score	0.78861448

Figure-7-:ROC for K-NN



5.2. Classification Trees (CART)

Essentially, the model tells us that if the last call duration (in seconds), is more than cutoff threshold value 225.50 (3.5 minutes) and if the cut off for outcome of previous marketing campaign (Poutcome_Success) is more than cutoff threshold 0.5 then the customer will subscribe to term deposit. Further, if the last call duration (in seconds) is more than 457.50(7.6 minutes) then customer will subscribe to term deposit. Further if we go down, if the communication type is unknown (Contact_Unkown) and the cut off threshold value is more than 0.5 then the customer will not subscribe to the term deposit, else yes. One more level down tell us that if a customer already has a housing loan with cut off threshold value less than 0.5 then the customer will subscribe to term deposit.

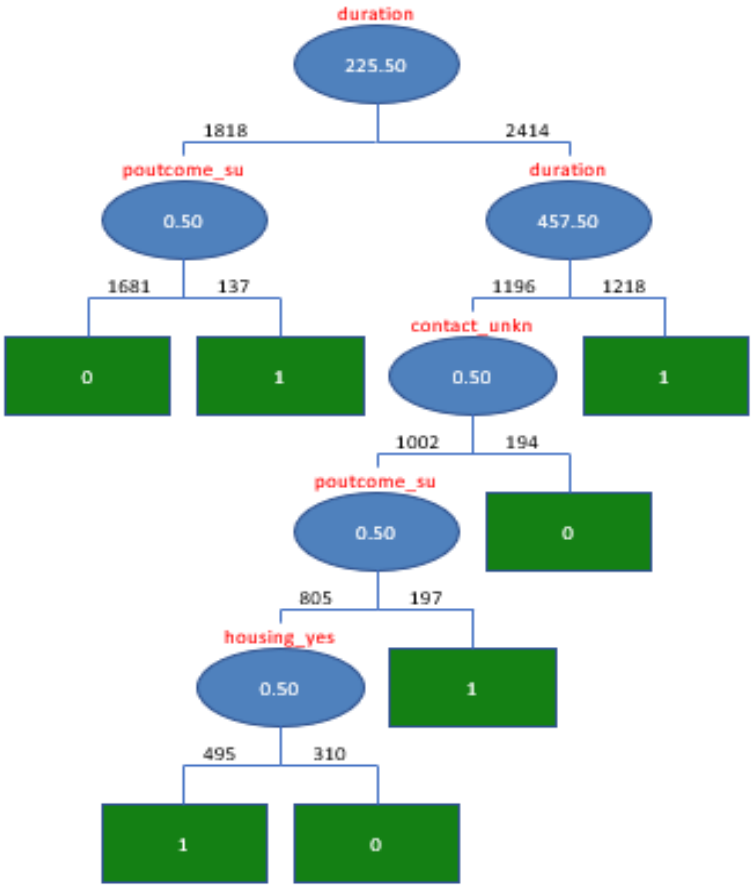


Figure-8:Best prune tree with six decision nodes with XLminer.

Figure-9:CART output

Validation Data scoring - Summary Report (Using Best Pruned Tree)

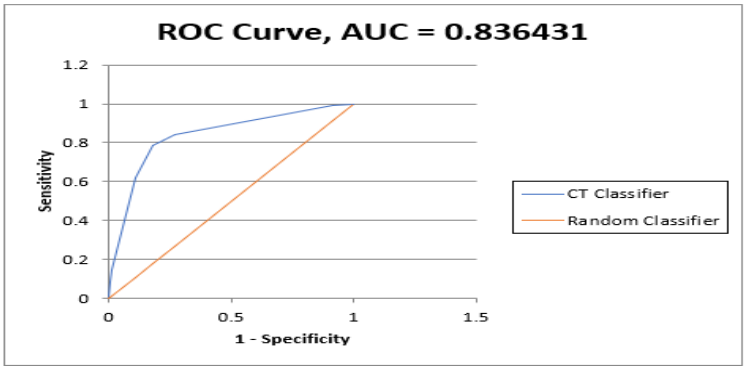
Cutoff probability value for success (UPDATABLE)	0.25	Updating the value here will NOT update value in detailed report

Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	1792	343
0	565	1532

Error Report			
Class	# Cases	# Errors	% Error
1	2135	343	16.06557
0	2097	565	26.94325
Overall	4232	908	21.45558

Performance	
Success Class	1
Precision	0.760289
Recall (Sensitivity)	0.839344
Specificity	0.730567
F1-Score	0.797863

Figure -10: CART ROC Curve



We are classifying the approximately 83% of our customers correctly who are subscribing to term deposit. The overall error rate for CART model which is 21.45% is decent for our data.
The area under the curve for ROC is 0.83 which is tending towards 1 but not the perfect curve.

5.3 Logistic Regression (LR)

The following figures illustrates the performance of our logistic regression model using forward selection, backward elimination and stepwise selection. As we mentioned above, the model developed using backward elimination and stepwise selection is the same, the only difference is the order of the variable selected.

Figure-11: Logistic Regression Performance Summary - Backward elimination and Stepwise Selection

Training Data Scoring - Summary Report

Cutoff probability value for success (UPDATABLE)	0.25

Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	3003	151
0	1285	1907

Error Report

Class	# Cases	# Errors	% Error
1	3154	151	4.79
0	3192	1285	40.26
Overall	6346	1436	22.63

Performance

Success Class	1
Precision	0.7003
Recall (Sensitivity)	0.9521
Specificity	0.5974
F1-Score	0.8070

Validation Data Scoring - Summary Report

Cutoff probability value for success (UPDATABLE)	0.25

Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	2028	107
0	867	1230

Error Report

Class	# Cases	# Errors	% Error
1	2135	107	5.01
0	2097	867	41.34
Overall	4232	974	23.02

Performance

Success Class	1
Precision	0.7005
Recall (Sensitivity)	0.9499
Specificity	0.5866
F1-Score	0.8064

Figure-12: Logistic Regression: ROC Curve - Backward elimination and Stepwise Selection

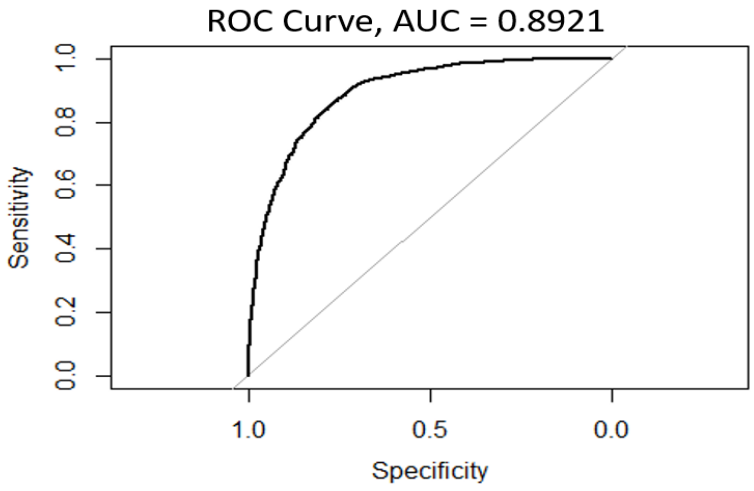


Figure-13 : Logistic Regression Performance Summary - Forward Selection

Training Data Scoring - Summary Report

Cutoff probability value for success (UPDATABLE)	0.25

Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	3005	149
0	1278	1914

Error Report

Class	# Cases	# Errors	% Error
1	3154	149	4.72
0	3192	1278	40.04
Overall	6346	1427	22.49

Performance

Success Class	1
Precision	0.7016
Recall (Sensitivity)	0.9528
Specificity	0.5996
F1-Score	0.8081

Validation Data Scoring - Summary Report

Cutoff probability value for success (UPDATABLE)	0.25

Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	2031	104
0	864	1233

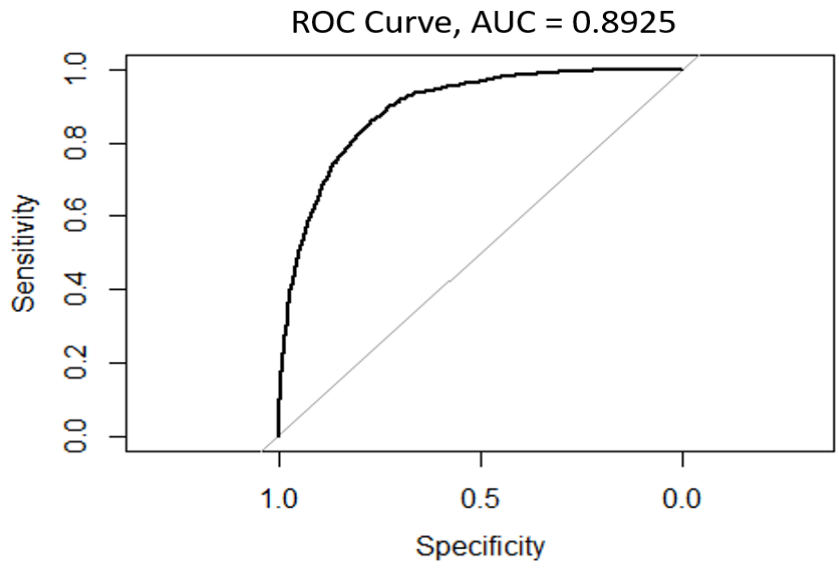
Error Report

Class	# Cases	# Errors	% Error
1	2135	104	4.87
0	2097	864	41.20
Overall	4232	968	22.87

Performance

Success Class	1
Precision	0.7016
Recall (Sensitivity)	0.9513
Specificity	0.5880
F1-Score	0.8076

Figure-14 : Logistic Regression: ROC Curve - Forward Selection



Comparing the training data overall error rate and validation data overall error rate in **Figure-12** and **Figure-13**, we can see that the validation data overall error rates increase slightly than those

of the training data. This increase is normal for validation data, and here we can tell there is no overfitting for the model.

Observing the overall error rate and sensitivity results from the three variable selection methods, we see that the forward selection method gives us the least overall error rate of 22.87 % and a higher sensitivity of 95.13 % than the other two methods. Thus, we rely on the forward selection method for logistic regression model formation. The receiver operating characteristic curve, i.e. ROC curve, also confirms our choice.

Figure-15: Logistic regression model under forward selection

Regression Model

Input Variables	Coefficient	Std. Error	P-Value	Odds	CI Lower	CI Upper
Intercept	-0.9214	0.247024	4.42684E-05	0.36464	0.224697	0.591741
duration	0.0055	0.000168	4.0515E-238	1.00555	1.005221	1.005884
poutcome_success	2.144	0.181037	8.47295E-33	8.6658	6.0773	12.3569
education_tertiary	0.2695	0.080144	0.000288588	1.3371	1.1428	1.5646
housing_yes	-0.9457	0.076737	2.90752E-35	0.3864	0.3324	0.4491
loan_yes	-0.6961	0.108167	2.81041E-10	0.5054	0.4088	0.6247
education_primary	-0.2894	0.117241	0.018626105	0.7589	0.603114	0.955
contact_unknown	-1.5080	0.11573	1.82565E-38	0.2229	0.1776	0.2796
Q3	-0.2768	0.091891	0.002860032	0.7602	0.6349	0.9103
marital_single	0.3343	0.086458	6.60358E-05	1.412	1.1919	1.6727
poutcome_unknown	-0.3653	0.167833	0.136411848	0.7788	0.560515	1.0822
job_Other	0.454	0.110981	5.85874E-05	1.562	1.2567	1.9416
Q4	0.3584	0.115493	0.001579478	1.4404	1.148617	1.8063
Q1	0.2832	0.111626	0.010419872	1.331	1.069462	1.6565
balance	0.00002	1.08E-05	0.055593939	1	1	1
poutcome_other	0.316	0.171691	0.076149623	1.3559	0.9685	1.8984
job_Unknown	0.9165	0.522473	0.117522997	2.2655	0.8137	6.3081

positive effect on outcome

negative effect on outcome

Residual DF	6319
Residual Dev.	5393.21336
Multiple R^2	0.38693954

Unlike multiple linear regression, the above model can NOT be interpreted as a linear equation of the predictors because here we have a categorical outcome(Yes/No). So we interpret this model as a 'logit' function which gives us the odds of the success class of outcome variable (client subscribing to a long term deposit) with respect to the predictor variables. Logit for above model can be explained as:

$$\begin{aligned}\log(odds) = {} & -.9214 + .0055duration + 2.144poutcome\_success \\ & + .2695education\_tertiary - .9457housing\_yes - .6961loan\_yes \\ & - .2894education\_primary - 1.5080contact\_unknown \\ & - .2768Q3 + .3343marital\_single - .3653poutcome\_unknown \\ & + .454job\_Other + .3584Q4 + .2832Q1 + .00002balance \\ & + .316poutcome\_other + .9165job\_Unknown\end{aligned}$$

Referring the regression model and the above logit equation, we can see that the predictors **duration of last contact call, successful outcome of previous campaign** and client **with educational qualification as tertiary** have positive coefficients and **the possession of housing loan, personal loan** and **client with educational qualification as primary have negative coefficients.**

Odds value greater than 1 or above depicts likelihood of the event happening.

Thus, we can make the following interpretations from the above model:

Positive effects on outcome (increase in the chances of success class):

Holding another predictor values constant,

1) The odds of a client subscribing to a long-term deposit will increase 0.55 times if the duration of last contact increase by 100 seconds.
2) The odds of a client subscribing to a long-term deposit are 8.66 if the outcome of previous campaign was a success.
3) The odds of a client subscribing to a long-term deposit are 1.337 if the educational qualification of client is tertiary.

Negative effects on outcome (decrease in the chances of success class):

Holding another predictor values constant,

1) The odds of a client subscribing to a long-term deposit are 0.3864 if the client has a housing loan.
2) The odds of a client subscribing to a long-term deposit are 0.5054 if the client has a personal loan.
3) The odds of a client subscribing to a long-term deposit are 0.7589 if the educational qualification of client is primary.

From the above analysis, we can state that a client with higher educational qualification will be more likely to subscribe to a long-term deposit. Also, the client already possessing a house loan or a personal loan will not prefer to subscribe to a long term deposit. With the bank point of view, bank should try to increase their contact call duration giving them detailed description of the benefits of the long-term deposit. This will increase the success factor of the campaign to encourage the clients to subscribe to the long term deposits.

6. Models Comparison

Figure-16: Comparison of Confusion Matrix for 3 Models

CONFUSION MATRIX								
k-NN			CART			Logistic Regression (Forward)		
	Predicted Class			Predicted Class			Predicted Class	
Actual Class	1	0	Actual Class	1	0	Actual Class	1	0
1	1844	251	1	1792	343	1	2031	104
0	759	1338	0	565	1532	0	864	1233

In figure-16, we are trying to compare the three-confusion matrix that we have received with three models. The Logistic Regression classifies maximum number of our '1' (one's) which is the

customer's subscribing to make term deposit. Also, it classifies the highest number of customer's who are not subscribing to long term deposit.

Figure-18: Performance comparison, cutoff= 0.25

Cutoff = .25	KNN	CART	LR
Overall Error Rate	23.87	21.46	22.87
Class "1" Error Rate	11.76	16.07	4.87
Sensitivity	0.88	0.83	0.95
Specificity	0.64	0.73	0.59

Figure-19: Performance comparison, cutoff = 0.5

Cutoff = .5	KNN	CART	LR
Overall Error Rate	22.09	19.80	19.14
Class "1" Error Rate	19.25	21.79	20.23
Sensitivity	0.81	0.78	0.80
Specificity	0.75	0.82	0.82

From above analysis, under the cutoff 0.25, now we have k-NN method with best k = 5, best prune tree with six decision nodes, and logistic regression model with forward selection. The **Figure-18** and **Figure-19** Illustrates the key performance measures of each method under cutoff 0.25 and cutoff 0.5. As we can see, from the overall error rate perspective, the CART gives the best result with an overall error rate of 21.46%, while LR have a slightly larger error rate of 22.87% and k-NN has the largest overall error rate. However, given that our task is to successfully identify potential customers who are more likely to subscribe the long-term deposit, Class "1" is more important here. In this case, LR gives the smallest Class "1" error rate of 4.87% and highest sensitivity among the three methods. It is noticeable that CART has a much larger Class "1" error rate than LR does but only has a slight advantage on the overall error rate. Compared the results under cutoff 0.25 and cutoff 0.5, the overall error rates slightly increase by using the lower cutoff threshold, which is normal for cutoff adjustment, however, the sensitivity of all three methods are improved and the Class "1" error rate is significantly improved, especially for the LR method.

Therefore, here we think cutoff 0.25 and the logistic regression are the best for this classification task.

The k-NN and CART are not ideal in our study is due to the small sample size. Both k-NN and CART are data-driven model, which requires large amount of data records. However, to construct a balanced dataset for this classification task, we must reduce the size of our dataset

by random sampling. As for logistic regression method, it not only does the classification task but also help us to profile the potential customers who are more likely to subscribe the product. And the bank could also use those characteristics to tailor its marks campaigns to those customers.

7. Conclusion

In this study, first we converted categorical variables into dummy variables and then did a random sampling to construct a balanced dataset. We used three different classification methods to classify the potential customers of a long-term bank deposit product. Since the bank would be more interested in identifying the customers who are more likely to successfully subscribe the product, here we use a lower cutoff 0.25 instead of 0.5. Based on the Class "1" error rate and sensitivity, we found out that logistic regression with forward selection has the best performance.

The performance of k-NN and CART are not that good in our case, probably because the relatively small size of our sample, which is a limitation of this study. In the field, there is likely to have a bigger balanced sample available. But the tradeoff would be the cost to collect and clean a bigger dataset and the time needed to process it. And when it comes to the best method, the main driver of the method selection principle is the task priority. If the task priority is to maintain a low overall error rate, the overall error rate of the validation dataset would be the primary index; if the task wants to identify the Class "1" as accurate as possible, and then Class "1" error rate and sensitivity would be the main consideration; if the task not only want to classify the output variable but also to identify some key characteristics of the subject, then CART and LR would provide a better perspective than k-NN dose. Meanwhile, cost-benefit and efficiency should always be in the considerations for best method selection, because it would be meaningless for any corporation to use a method which cost too much and take too long. The comparable advantage of data mining is relatively short-timed.

8. References

[1] S. Moro, P. Cortez & P. Rita (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems, 62, 22-31.*

[2] C. Vajiramedhin & A. Suebsing (2014). Feature Selection. *Applied Mathematical Sciences, Vol. 8, no. 104, 5667-5672.*

[3] S. Moro, R. Laureano and P. Cortez(2011). Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. 117-121.

[4] Shmueli, Patel and Bruce, Data Mining for Business Analytics: Concepts, Techniques, and Applications with XLMiner, John Wiley & Sons, 3rd edition.

[5] Dr. Yinfei Kong slides on Titanium Portal.

9. Appendix

9.1. Data source: https://archive.ics.uci.edu/ml/datasets/bank+marketing

9.2. First 20 records of the data

Figure: Screenshot of the first 20 records after reducing categories of job and month variables.

age	job_red	marital	education	default	balance	housing	loan	contact	day	month_red	duration	campaign	pdays	previous	poutcome	y
49	3	married	secondary	no	4110	no	no	unknown	17	2	50	2	-1	0	unknown	no
29	1	single	secondary	no	788	yes	no	cellular	18	3	264	2	-1	0	unknown	no
35	1	single	tertiary	no	385	yes	no	cellular	29	3	168	2	-1	0	unknown	no
26	1	single	tertiary	no	16563	yes	no	cellular	18	2	245	1	-1	0	unknown	no
35	1	single	secondary	no	1708	yes	no	unknown	2	2	110	3	-1	0	unknown	no
33	2	single	primary	no	165	yes	no	cellular	15	2	139	2	-1	0	unknown	no
62	1	married	tertiary	no	10373	no	no	cellular	12	3	344	3	-1	0	unknown	no
29	1	single	tertiary	no	8	no	no	cellular	28	1	468	1	-1	0	unknown	no
30	1	single	tertiary	no	2766	yes	yes	cellular	20	4	244	1	-1	0	unknown	no
25	1	single	tertiary	no	1270	yes	no	unknown	13	2	124	2	-1	0	unknown	no
30	1	single	tertiary	no	234	yes	no	unknown	3	2	242	2	-1	0	unknown	no
36	1	single	primary	no	235	yes	no	unknown	29	2	141	9	-1	0	unknown	no
47	1	married	secondary	no	1803	no	no	cellular	30	1	188	2	-1	0	unknown	no
54	2	married	primary	no	167	yes	no	cellular	22	3	106	2	-1	0	unknown	no
38	1	married	tertiary	no	5432	yes	yes	cellular	17	2	707	1	149	2	failure	no
44	1	divorced	tertiary	no	0	no	no	cellular	13	3	169	2	-1	0	unknown	no
34	1	married	secondary	no	314	yes	yes	telephone	5	1	114	3	-1	0	unknown	no
48	1	divorced	secondary	no	1033	no	yes	unknown	22	3	11	1	-1	0	unknown	no
57	2	married	primary	no	688	no	no	cellular	21	3	67	2	-1	0	unknown	no
40	1	married	secondary	no	1060	yes	no	cellular	17	2	14	3	-1	0	unknown	no
22	3	single	tertiary	no	1161	no	yes	cellular	16	2	119	1	-1	0	unknown	no

9.3 R codes

9.3.1 Data preprocessing

```
#random sampling#
dat = read.csv('DC.csv', stringsAsFactors=T, head=T)
dat0 = dat[which(dat$y_yes == 0), ] #no of 0s i.e. no outcome values in y
dat1 = dat[which(dat$y_yes == 1), ]#no of 1s i.e. yes outcome values in y
set.seed(1)
ind = sample(1:nrow(dat0), nrow(dat1)) #taking sample by referring total no of 1s
dat00 = dat0[ind, ]#seperating the equal no of 0s with 1s
datnew = rbind.data.frame(dat00, dat1)#combining separated 0s and original no of 1s
write.csv(datnew, file='datanew.csv', row.names=F)
```

9.3.2. Data partition

```
rm(list=ls());gc()
setwd('/Users/Owner/Desktop/Fall 2017/ISDS 574 Data Mining/project/')
dat = read.csv('SampleData1.0.csv', head=T, stringsAsFactors=F, na.strings='')

##1. Take 60% of data randomly as training##
set.seed(1)
id.train = sample(1:nrow(dat), nrow(dat)*.6) # ncol() gives number of columns
id.test = setdiff(1:nrow(dat), id.train) # setdiff gives the set difference

##2. Prepare data for XLMiner that have same partition##
ind_XLMiner = rep(NA, nrow(dat))
ind_XLMiner[id.train] = 'T'
ind_XLMiner[id.test] = 'V'
```

```
dat1 = cbind(dat, ind_XLMiner)
write.csv(dat1, file = 'SampleData2.0.csv', row.names=F, na='')
```

9.3.3. k-Nearest-Neighbors (kNN)

```
# remove all the variables stored in the environment
rm(list = ls())gc()
# Set the location
setwd('/Users/Owner/Desktop/Fall 2017/ISDS 574/Final Project')
# Read the file
dat = read.csv('SampleData1.csv', header = TRUE, stringsAsFactors = TRUE)

# normalize the data except the outcome variable (dat[1]) and
# the partition indicator variable (dat[28])
normaldata = as.data.frame(scale(dat[2:27], center=TRUE, scale=TRUE))
# combine the normalized data with outcome variable and partition indicator
finnormaldata = as.data.frame(c(dat[1], normaldata, dat[28]))
# use the indicator variable to build the train data
train = finnormaldata[finnormaldata$ind_XLMiner=='T',]
# remove the indicator variable from the train data
train = subset(train, select = -c(ind_XLMiner))
# specify the outcome variable in y.train
y.train = train[,1]
# use the indicator variable to build the test data
test = finnormaldata[finnormaldata$ind_XLMiner=='V',]

# remove the indicator variable from the test data
test = subset(test, select = -c(ind_XLMiner))

# specify the outcome variable in y.test
y.test = test[,1]

require(class)
require(caret)

# build a dich function that will take in a probability vector and
# use a cutoff to build the outcome as 1 if probability is above
# the specified cutoff
dich = function(x, ct = .5) {
  out = rep(0, length(x))
  out[x > ct] = 1
  out
```

```
}
# build a function to find the best k that produces the least error
knn.bestK = function(train, test, y.train, y.test, k.max = 20, ctoff = 0.5) {
  k.grid = seq(1, k.max)
  error = rep(NA, length(k.grid))
  fun.tmp = function(x) {
    y.prob = attributes(knn(train, test, y.train, y.test, k = x,prob=T))$prob
    y.hat = dich(y.prob, ctoff)
    return(sum(y.hat != y.test))
  }
  ## create a temporary function (fun.tmp) that we want to apply to each value in
k.grid
  error = unlist(lapply(k.grid, fun.tmp)) / length(y.test)
out = list(k.optimal = k.grid[which.min(error)], error.min = min(error))
  return(out)
}
# find out the best k by using knn.bestK function
knn.bestK(train[2:27], test[2:27], y.train, y.test, 20, 0.25)
```

9.3.4. Classification Trees (CART)

```
list=ls()); gc()
library(rpart)

dat = read.csv('SampleData1.0.csv', stringsAsFactors=T, head=T)
colnames(dat)
head(dat)

# Classification Tree with rpart
# grow tree
fit = rpart(y_yes ~ ., method="class", data=dat, cp = 1e-2, minsplit=21)

printcp(fit) # display the results
plotcp(fit) # visualize cross-validation results
summary(fit) # detailed summary of splits

# plot tree
plot(fit, uniform=T, main="Classification Tree for Loan Acceptance data")
text(fit, use.n=T, all=TRUE, cex=.8)

# prune the tree
```

```
pfit = prune(fit, cp = fit$cptable[which.min(fit$cptable[,"xerror"]),"CP"])

# plot the pruned tree
plot(pfit, uniform=T, main="Pruned Classification Tree for Loan Acceptance Data")
text(pfit, use.n=T, all=T, cex=.8)
```

9.3.5. Logistic Regression (LR)

```
rm(list=ls());gc()
setwd('/Users/Owner/Desktop/Fall 2017/ISDS 574 Data Mining/project/')
dat = read.csv('SampleData2.0.csv', head=T, stringsAsFactors=F, na.strings='')

##1. Building min.model and max.model##
dat.train = dat[id.train,]
dat.test = dat[id.test,]
min.model = glm(y_yes ~ 1, data = dat.train, family = 'binomial')
max.model = glm(y_yes ~ ., data = dat.train, family = 'binomial')
max.formula = formula(max.model)

##2. Building function for classification task##
dichotomize = function(x, cutoff=.5) {
  out = rep(0, length(x))
  out[x > cutoff] = 1
  out
}

##3. Forward selection##
objf = step(min.model, scope=list(lower=min.model, upper=max.model), direction='forward')
summary(objf)
yhatf = predict(objf, newdata = dat.test, type='response')
hist(yhatf)

yhatf.class1 = dichotomize(yhatf, .5)
sum(yhatf.class1 != dat.test$y_yes)/length(id.test) ##misclassification error rate for cutoff 0.5

yhatf.class2 = dichotomize1(yhatf, .25)
sum(yhatf.class2 != dat.test$y_yes)/length(id.test) ##misclassification error rate for cutoff 0.25

#Obtain confusion matrix, error rate, sensitivity, specitivity#
install.packages('caret')
require(caret)
confusionMatrix(yhatf.class2, dat.test$y_yes, positive= "1") ##confusion matrix
```

```
#Obtain ROC Curve#
install.packages('pROC')
require(pROC)
roc(dat.test$y_yes, yhatf)
plot.roc(dat.test$y_yes, yhatf)

##4. Backward elimination#
objb = step(max.model, scope=list(lower=min.model, upper=max.model), direction='backward')
summary(objb)

yhatb = predict(objb, newdata = dat.test, type='response')
hist(yhatb)

yhatb.class = dichotomize(yhatb, .25)
sum(yhatb.class != dat.test$y_yes)/length(id.test) ##misclassification error rate for cutoff 0.25

#Obtain confusion matrix, error rate, sensitivity, specitivity#
confusionMatrix(yhatb.class, dat.test$y_yes, positive= "1") ##confusion matrix

#Obtain ROC Curve#
roc(dat.test$y_yes, yhatb)
plot.roc(dat.test$y_yes, yhatb)

##5. Stepwise selection##
objsw = step(min.model, scope=list(lower=min.model, upper=max.model), direction='both')
summary(objsw)

yhatsw = predict(objsw, newdata = dat.test, type='response')
hist(yhatsw)

yhatsw.class = dichotomize(yhatsw, .25)
sum(yhatsw.class != dat.test$y_yes)/length(id.test) ##misclassification error rate for cutoff 0.25

#Obtain confusion matrix, error rate, sensitivity, specitivity#
confusionMatrix(yhatsw.class, dat.test$y_yes, positive= "1") ##confusion matrix

#Obtain ROC Curve#
roc(dat.test$y_yes, yhatsw)
plot.roc(dat.test$y_yes, yhatsw)
```

9.3.6. Prune Tree from R code