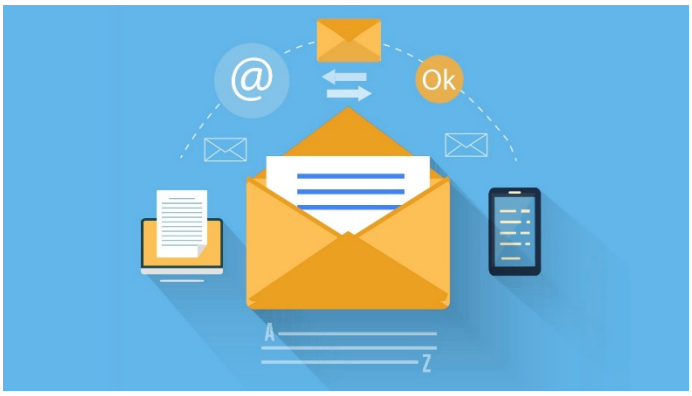
Solo 3: Customer Segmentation and Target Marketing



Name: Young, Brent

MSDS 450 Section #: 55

Quarter: Winter 2019

Introduction

Problem

Founded in 1985, XYZ is a Multi-Channel Retailer of consumer electronics and appliances with 200 stores nationwide and a direct-mail business that distributes millions of catalogs each year. As a result, the purpose of Solo 3 Assignment is to use XYZ's data to develop a target marketing response model that predicts responses to the most recent (16th) mail campaign and then use it to tell XYZ which customers it should send a catalog to in its next (17th) mail campaign so that the net returns (total returns – mailing costs) can be maximized. In order to achieve this, we will split the data into a training/validation set, develop different predictive models on the training set using initial binary responses (likely buyers vs. non-buyers; dependent variable) from XYZ's most recent mail campaign and specific customer information (independent variables), and evaluate the models on the validation set using statistical evaluation criteria. Upon identifying the best predictive model and using the model results/other information available, XYZ customers that should be sent a catalog will be identified using the financial evaluation criteria and the expected net revenue (e.g. net profit – mailing costs) from mailing to them will be communicated. Customers that XYZ didn't target in its most recent campaign (e.g., hold-out set) will then be identified based on who we predicted will produce net revenue so that XYZ can mail catalogs to them and an estimate of expected net revenue from mailing to these customers will be provided. We will then compare the expected net revenue from the most preferred model versus the actual revenue that resulted from XYZ's 16th campaign and will provide model interpretations from a marketing perspective for the most preferred model. To conclude the analysis, we will tell XYZ how they can apply the final model to new customers that are yet to be acquired in their next mail campaign. An approach to assessing predicted responses to XYZ's next campaign by targeting selected customers will then be designed and communicated.

Significance

The problem is significant and interesting because according to XYZ's recent mailing records, it costs XYZ $3.00 to produce and mail a catalog to each customer. However, XYZ earns 10% profit on items it sells through its catalogs (excludes costs of the catalog and mailing).

Exploratory Data Analysis

Structure and Description of Training, Validation Datasets

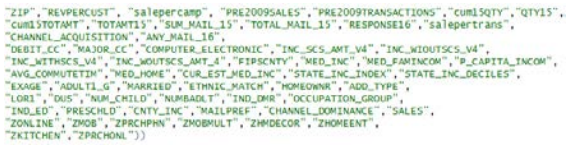
The XYZ customer data is stored in three binary data files called XYZ_customer_data.RData, XYZ_item_data.RData, and XYZ_mail_data.RData and integrated within a data file called XYZ_complete_customer_data_frame.RData, which consists of complete customer profile data. The structure of the entire dataset includes 30779 rows or customers and 554 columns (e.g., variables such as sales/purchase history, location/zip code, transaction history, demographics, etc.). Additionally, there are 345 character, 161 double, and 48 integer variable types with RESPONSE16 representing our classification response variable (1 = Bought, 0 = Did Not Buy) for the 16^{th} campaign. In fact, the data for the 16^{th} campaign shows that 2592 people responded and 28187 did not respond. Furthermore, 14992 people were targeted (received a mailer), while 15857 people were not targeted (did not receive a mailer). The data also shows that 14705 people did not respond and weren't targeted, 13483 people did not respond and were targeted, 1152 people did respond and weren't targeted, and 1440 people did respond and were targeted. The data was then split into a 70/30: train and validation data sets. For instance, there are 10446 training observations (70%) and 4476 validation observations (30%) for the people who were sent mailers in the 16^{th} campaign (e.g., ANY_MAIL_16 > 0). The training set will be used to fit the models and the validation set will be used to estimate prediction error for model selection. The best model will then be applied to subdat2 (e.g., all data in subset) for financial evaluation purposes.

Feature Engineering & High Level Summary Statistics

Through feature engineering, we created 8 new numeric variables: Sum of Total Sales By Customer for Previous Campaigns (cum15TOTAMT), Revenue Per Customer (REVPERCUST), Total Sales Before 2009 (PRE2009SALES), Total # of Transactions Before 2009 (PRE2009TRANSACTIONS), Sum of # of Items Ordered for Previous Campaigns (cum15QTY), Sales Per Transaction (salepertrans), and Sum of Total Sales for 15th Campaign (salepercamp). Here's a summary on how these variables were calculated: cum15TOTAMT = TOTAMT0 + TOTAMT0 +…+ TOTAMT15; REVPERCUST =

cum15TOTAMT*1.10 - TOTAL_MAIL_15*3.00, where TOTAL_MAIL_15 is the total number of mailers received by the end of the 15th campaign; PRE2009SALES = LTD_SALES - YTD_SALES_2009; PRE2009TRANSACTIONS = LTD_TRANSACTIONS - YTD_TRANSACTIONS_2009; cum15QTY = QTY0 + QTY1 +…+ QTY15; salepertrans =PRE2009SALES/PRE2009TRANSACTIONS; and salepercamp = cum15TOTAMT/TOTAL_MAIL_15. Overall, quick summary statistics on the *entire dataset* reveal that average total sales per customer before 2009 is $979.22, with the min = 0 and max = 94350; average revenue per customer for previous campaigns is $192.67; average transactions per customer before 2009 is 3.8; average total sales per customer for previous campaigns is $187; average quantity purchased for previous campaigns is 1.9; average sales per transaction is $233, and average cumulative sales up to the 15th campaign is $36. *Note: I also tried using log transformations on some of the income related variables, but they errored out when I applied it to my models due to the fact that the summary statistics for these variables showed "inf".*

Subset 'Important' Predictors & Variable Types



The image on the right shows a subset of 56 predictors (includes RESPONSE16) that will be used for EDA and model building purposes. The predictors were chosen based on domain expertise and includes variables such as sales/purchase history, buyer status, payment type, location/zip code, median income, demographics, occupation, interests, etc. We then picked only the people who were sent mailers in Campaign 16 (e.g., ANY_MAIL_16 > 0). We then changed the following variables to factors: RESPONSE16, ZIP, CHANNEL_ACQUISITION, DEBIT_CC, MAJOR_CC, COMPUTER_ELECTRONIC, INC_WIOUTSCS_V4, INC_WITHSCS_V4, FIPSCNTY, ADULT1_G, MARRIED, ETHNIC_MATCH, HOMEOWNR, ADD_TYPE, DUS, IND_DMR, OCCUPATION_GROUP, IND_ED, PRESCHLD, MAILPREF, CHANNEL_DOMINANCE, SALES, ZONLINE, ZMOB, ZPRCHPHN, ZMOBMULT, ZHMDECOR, ZHOMEENT, ZKITCHEN, and ZPRCHONL. Additionally, EXAGE, LOR1, NUM_CHILD, SUM_MAIL_15, TOTAL_MAIL_15, REVPERCUST, and salepercamp were changed to numeric variables. ANY_MAIL was then removed from the subset.

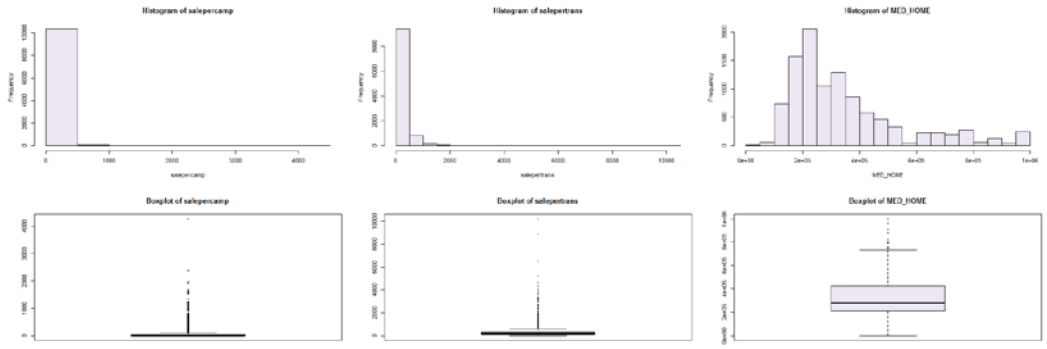
Data Cleaning & Imputation

After conducting summary statistics on the dataset and converting the blanks to NA, the results showed that there are 44133 NA's. The histogram in Appendix 1 shows the total number of missing values for all the variables in the dataset using the VIM package. For instance, the following variables had missing values: SALES (5191), DEBIT_CC (4931), MAJOR_CC (4931), COMPUTER_ELECTRONIC (4931), salepercamp (2487), EXAGE (2097), salepertrans (1294), CHANNEL_DOMINANCE (1274), *ETHNIC_MATCH (1225)*, *HOMEOWNR, ADD_TYPE, IND_*DMR*, MAILPREF*, *ZONLINE, ZMOB, ZPRCHPHN, ZMOBMULT, ZHMDECOR, ZHOMEENT, ZKITCHEN, ZPRCHONL*, NUM_CHILD (1071), and ZIP (1). *Note: Italicized variables each had 1225 missing values.* To address the missing values, I decided to fill in all the factor variables (blue variables), except ETHNIC_MATCH with the majority value (e.g., either with 60091, U, C, Y, S, N depending on the variable). The NA's for ETHNIC_MATCH were filled with N to represent "No" and the NA's for SALES were filled with U to represent "Unknown". In regards to the numeric variables (green variables), the NA's and U for EXAGE were filled with the median age, while the NA's and *inf (salepercamp only)* for salepercamp, salepertrans, and NUM_CHILD were filled with 0.

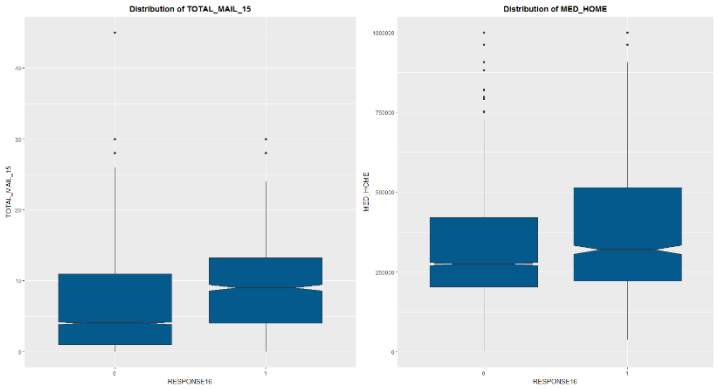
EDA for Classification Models

Descriptive Statistics & Univariate Plots of Numeric Variables

Appendix 2, shows summary statistics of the numeric variables in the training dataset so that we can check for missing values, outliers, distributions, etc. For instance, the data shows that the mean for salespercamp is 37.83, salepertrans is 251.7, REVPERCUST is 265.8, PRE2009SALES is 1276, PRE2009TRANSACTIONS is 4.492, MED_HOME is 345792, cum15QTY is 2.517, and INC_WOUTSCS_AMT_4 is 140.6. Histograms and boxplots also revealed that there are outliers for all the numeric variables, except SUM_MAIL_15, INC_SCS_AMT_V4, and INC_WOUTSCS_AMT_4. However, I decided not to handle/truncate the outliers because it did not improve my models much when I tested them on the validation dataset. Furthermore, majority of the numeric variables had either a right or left skew, except AVG_COMMUTETIM and EXAGE which have a normal distribution. The histograms and boxplots of salespercamp, salespertrans, and MED_HOME below, illustrates an example of the right skew that was mentioned.



Multivariate Plots

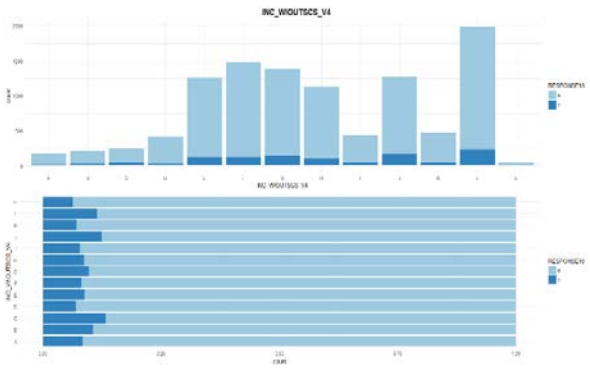


The figure to the right shows notched boxplots of TOTAL_MAIL_15 (left) and MED_HOME (right). vs. RESPONSE16, so that we can compare the median differences and variability between the numeric variable and RESPONSE16. The notch displays a confidence interval around the median which is normally based on the median +/- 1.58*IQR/sqrt(n), which allows us to visually compare if the medians differ. The results show that as TOTAL_MAIL_15 and MED_HOME increase, the more likely a customer is to buy. Overall, this provides evidence that these variables are potentially strong predictors to include in our models since the median difference between whether a customer bought or did not buy (0 = No, 1 = Yes) is wide. The notched boxplots confirmed this as well. For instance, since the notches of the two boxes do not overlap, there is strong evidence that the medians differ. I also conducted additional notched boxplots of RESPONSE16 (x-axis) vs. the other numeric variables (y-axis) and found that salepertrans, salespercamp, REVPERCUST, PRE2009SALES, PRE2009TRANSACTIONS, cum15QTY, cum15TOTAMT, INC_SCS_AMT_V4, INC_WOUTSCS_AMT_4, MED_INC, MED_FAMINCOM, P_CAPITA_INCOM, CUR_EST_MED_INC, STATE_INC_INDEX, and CNTY_INC have median differences between RESPONSE16 as well, while the rest did not and were often removed from my models. Interestingly, the variable importance function in the Caret package confirmed that majority of these variables are "important" as well.

Correlation Matrix

The correlation matrix in Appendix 3 allows us to see which variables may be correlated with each other so that we can gleam interesting insights. The plot shows that the sales versus transaction history variables have strong positive correlations with each other, while many of the income related variables have strong positive correlations with each other as well. For instance, as CUM15QTY increases, REVPERCUST also increases (0.78). Similarly, as STATE_INC_INDEX increases, MED_HOME also increases (0.94). This indicates that some of the variables can be removed since they may be another way of measuring the same response when predicting customer responses.

Qualitative Variables



Observations: The stacked bar chart (top) on the right shows the various counts by RESPONSE16 for INC_WIOUTSCS_V4 for each group. Additionally, the stacked bar chart (bottom) shows RESPONSE16 on the x-axis (light blue = 0, dark blue = 1) and various groups on the y-axis as a proportion. The data shows that those who fall in different groups within INC_WIOUTSCS_V4 are more willing to buy compared to others. As a result, this provides evidence of possibly including INC_WIOUTSCS_V4 in our models. We also did a similar analysis with the other factor variables and

found differences between groups for the following variables: ZIP, INC_WITHSCS_V4, ZPRCHONL, CHANNEL_ACQUISITION, ADULT1_G, CHANNEL_DOMINANCE, ZHMDECOR, ZONLINE, and ADD_TYPE, while the others either had limited variation and/or the sample size of one group dominated the other.

Modeling Strategy

Given that the goal of this assignment is to develop a model that predicts responses to the last (the 16th) mail campaign using the data we have available, and then use it to tell XYZ which customers it should send a catalog to in its next (17th) mail campaign so that net returns are maximized, I began my analysis using logistic regression with stepAIC to serve as an initial baseline prior to conducting more sophisticated modeling techniques. After building the logistic regression model, I then moved to linear discriminant analysis (LDA) and concluded with random forest. The next section provides a summary of my results for each modeling technique.

Formulation of Models

Classification Models

Logistic Regression: Logistic regression models the probability that the response variable belongs to a specific category and assumes a linear decision boundary (James, Witten, Hastie, & Tibshirani, 2013). For instance, it models the probabilities of the K classes using linear functions in x, while also ensuring that they sum to 1 and remain in-between 0 and 1 (Hastie, Tibshirani, & Friedman, 2009). This is accomplished using the logistic function and maximum likelihood, which is used to fit the model (James, et al., 2013). As a result, using the glm function, I produced a logistic regression model of RESPONSE16 ~ PRE2009TRANSACTIONS + TOTAL_MAIL_15 + QTY15 + CHANNEL_ACQUISITION + salepercamp + ADULT1_G + PRE2009SALES + MED_HOME + ZPRCHONL + cum15QTY + CHANNEL_DOMINANCE + INC_WOUTSCS_AMT_4 + HOMEOWNR + AVG_COMMUTETIM + ZHMDECOR. These variables were chosen using stepwise regression (stepAIC). The hoslem.test confirmed that the model is statistically significant given the p-value of < 2.2e-16. The Analysis of Deviance table (e.g., difference between the null deviance and the residual deviance of adding each predictor individually; the wider the gap, the better) and varimp showed that PRE2009TRANSACTIONS, impacted the model the most. In regards to the coefficients of the model, variables such as PRE2009TRANSACTIONS, which has a positive coefficient make intuitive sense and were statistically significant. For instance, as PRE2009TRANSACTIONS increases, the more likely a customer is likely to buy (vice versa). The model produced the following performance metrics on the training dataset: AIC: 6100.637, BIC: 6267.478 and the following AUC: 0.7151599 and AUPRC: 0.2142101 on the validation dataset. *Note: Given a randomly chosen observation from the positive class (1=buy) and a randomly chosen observation from the negative class (0 = did not buy), the area under the curve (AUC) is the probability that the evaluated model will assign a higher score to the positive class compared to the negative class. In other words, the AUC is the probability that the classifier will rank a randomly selected individual from the positive class higher than a randomly chosen individual from the negative class (i.e., the average value of sensitivity of a classifier for all possible values of specificity and vice versa) (Fawcett, 2006).* However, it's important to note that using ROC AUC for imbalanced classes can be misleading and that calculating area under the precision-recall curve (AUPRC) is recommended instead since they are not impacted by the true negatives (Saito and Rehmsmeier, 2015). Appendix 4 shows a histogram of probabilities on the validation set. Majority of the probabilities fall in-between 0 to 0.2 ("L-shaped" histograms). See Appendix 4 for ROC and Precision-Recall Curve as well.

Linear Discriminant Analysis: LDA is very similar in form to logistic regression (distributions are assumed to be normal), except it models the distribution of the predictors separately in each of the response classes and then applies Bayes theorem (James, et al., 2013). This model also uses Gaussian densities, which arises when we assume that the classes have a common covariance matrix, and assumes a linear decision boundary (Hastie, et al., 2009). Using the lda function, I then produced a linear discriminant analysis model of RESPONSE16 ~ PRE2009TRANSACTIONS + TOTAL_MAIL_15 + QTY15 + CHANNEL_ACQUISITION + salepercamp + ADULT1_G + PRE2009SALES + MED_HOME + ZPRCHONL + cum15QTY + CHANNEL_DOMINANCE + INC_WOUTSCS_AMT_4 + HOMEOWNR + AVG_COMMUTETIM + ZHMDECOR. These variables were chosen using the same variables from the logistic regression model. The model produced the following AUC: 0.7193693 and AUPRC: 0.2198539 on the validation dataset and performed similarly to logistic regression. Appendix 5 shows a histogram of probabilities on the validation set. Majority of the probabilities fall in-between 0 to 0.2 ("L-shaped" histograms). See Appendix 5 for ROC and Precision-Recall Curve as well.

Random Forest with SMOTE: One of the primary implications of rare outcomes for developing accurate and useful models is that some predictive models tend to produce unsatisfactory classifiers *(i.e., the model predicts the majority class since the minority class is treated as noise and is often ignored) (*Mukherjee, 2017). This results in a high probability of misclassification of the minority class as compared to the majority class, a key limitation that was seen in both the logistic regression and LDA models given the prevalence of 0.0965 (indicates that only about 10% of customers bought items) *(*Mukherjee, 2017). As a result, I decided to use SMOTE (Synthetic Minority Over-sampling Technique), which is a statistical technique that generates synthetic samples from the minority class and oversamples the dataset in order to increase the number of rare cases and balance the dataset prior to using randomForest (Brownlee, 2015). Here are the parameters that I used: SMOTE(RESPONSE16 ~., train, perc.over = 100, k = 5, perc.under = 200). Random forest provides an improvement over bagged trees by incorporating a small tweak that decorrelates the trees and then averages them (e.g., forces each split to only consider a subset of predictors and will not consider strong predictors so that other predictors will have more of a chance (James, et al., 2013)). As a result, using the randomForest function (ntree=100), a random forest with SMOTE model was produced using RESPONSE16 ~ PRE2009TRANSACTIONS + TOTAL_MAIL_15 + PRE2009SALES + MED_HOME + cum15QTY + INC_WIOUTSCS_V4. These variables were chosen using the variable importance function (which measures prediction strength) and choosing the top 20 predictors (see Appendix 6) based on MeanDecreaseAccuracy and then selecting the variables that overlapped with the logistic regression model (stepAIC) *(Note: INC_WIOUTSCS_V4 was used instead of INC_WOUTSCS_AMT_4 because it had a higher MeanDecreaseAccuracy).* Our EDA also validated that these variables are important as well. The model produced the following AUC: 0.6850298 and AUPRC: 0.1906895 on the validation dataset. Appendix 6 shows a histogram of the probabilities of the random forest with SMOTE model on the validation set. Majority of the probabilities fall near 0 to 0.3 and then steadily decline in a "staircase" fashion, which is a significant contrast to the "L-shaped" histograms that logistic regression and LDA produced. See Appendix 6 for ROC and Precision-Recall Curve as well.

Results: Performance/Accuracy of Classification Models on Validation Set

Logistic Regression

```
Confusion Matrix and Statistics

glm.pred    0    1
       0 4021  418
       1   23   14

               Accuracy : 0.9015
                 95% CI : (0.8924, 0.9101)
    No Information Rate : 0.9035
    P-Value [Acc > NIR] : 0.6865

                  Kappa : 0.0452
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.032407
            Specificity : 0.994313
         Pos Pred Value : 0.378378
         Neg Pred Value : 0.905835
             Prevalence : 0.096515
         Detection Rate : 0.003128
   Detection Prevalence : 0.008266
      Balanced Accuracy : 0.513360

       'Positive' Class : 1
```

LDA

```
Confusion Matrix and Statistics

lda.pred    0    1
       0 3991  406
       1   53   26

               Accuracy : 0.8975
                 95% CI : (0.8882, 0.9062)
    No Information Rate : 0.9035
    P-Value [Acc > NIR] : 0.9171

                  Kappa : 0.0741
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.060185
            Specificity : 0.986894
         Pos Pred Value : 0.329114
         Neg Pred Value : 0.907664
             Prevalence : 0.096515
         Detection Rate : 0.005809
   Detection Prevalence : 0.017650
      Balanced Accuracy : 0.523540

       'Positive' Class : 1
```

Random Forest w/ SMOTE

```
Confusion Matrix and Statistics

glm.pred    0    1
       0 3057  204
       1  987  228

               Accuracy : 0.7339
                 95% CI : (0.7207, 0.7468)
    No Information Rate : 0.9035
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1568
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.52778
            Specificity : 0.75593
         Pos Pred Value : 0.18765
         Neg Pred Value : 0.93744
             Prevalence : 0.09651
         Detection Rate : 0.05094
   Detection Prevalence : 0.27145
      Balanced Accuracy : 0.64186

       'Positive' Class : 1
```

Model Name	AUC	AUPRC
Logistic Regression	0.7151599	0.2142101
LDA	0.7193693	0.2198539
Random Forest w/ SMOTE	0.6850298	0.1906895

Observations: The confusion matrix and statistics above shows the performance/accuracy metrics and evaluation criteria for the following models in the validation dataset: logistic regression, LDA, and random forest with SMOTE. The results show that the random forest with SMOTE model produced the highest sensitivity (true positive rate), despite a high number of false positives, lowest specificity (true negative rate), and similar AUC and AUPRC compared to logistic regression and LDA. *Note: Using accuracy for an imbalanced dataset is misleading and was not used as a metric for model comparison.* However, this is a worthy tradeoff given that: (1) we are trying to predict likely buyers and (2) XYZ is a consumer electronics and appliances store that sells expensive items, so a $3.00 cost to produce and mail a catalog is very minimal, especially if it results in a high priced sale and more profit. For instance, in comparison to the logistic regression and LDA models, we can see the very low sensitivity metrics and "L-shaped" histograms of probabilities (see Appendix 4 & 5), which indicates that both models had a difficult time predicting the minority class (buyers). However, in an alternative scenario where the cost is high or potentially higher than a potential sale, minimizing false positives would be the point of emphasis. Given this justification, I applied the random forest with SMOTE model to subdat2 (e.g., all data in subset) for financial evaluation/summary purposes.

Financial Evaluation Part I

	Random Forest Model Cutoff Rules							
Predicted Probability (percentage)	65.00	50.00	40.00	30.00	20.00	15.00	10.00	5.00
XYZ Targeting with New Model								
Targeted Customers								
Number of Customers Targeted	2332	4003	5353	7090	9416	10706	12120	13660
Average Revenue per Customer	699.7214	615.9711	537.7053	454.294	380.1277	350.9264	321.3439	292.8959
Direct mail cost per Customer	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Ave. Revenue Minus Mail Cost per Customer	696.72	612.97	534.71	451.29	377.13	347.93	318.34	289.90
Profit with New Model	$1,624,754	$2,453,723	$2,862,277	$3,199,674	$3,551,034	$3,724,900	$3,858,328	$3,959,978
Profit Increase or Loss with New Model	($2,375,903)	($1,546,934)	($1,138,379)	($800,982)	($449,622)	($275,757)	($142,329)	($40,679)
Per Customer Profit Contribution or Loss	($159.22)	($103.67)	($76.29)	($53.68)	($30.13)	($18.48)	($9.54)	($2.73)
Number of Customers in Database	30,779	30,779	30,779	30,779	30,779	30,779	30,779	30,779
Estimated Profit Contribution/Lift of Targeting	($4,900,677)	($3,190,797)	($2,348,089)	($1,652,154)	($927,418)	($568,792)	($293,576)	($83,907)

Observations: The table above shows financial evaluation/summary data for the random forest with SMOTE model (e.g., ANY_MAIL_16 > 0) so that we can identify which customers should be sent a catalog, and what net revenue we expect will result from mailing to them. Here is how the calculations were computed: Average Revenue per Customer for All = mean(data_alldf$REVPERCUST); Targeted Customers with New Model = sum(data_alldf$prediction_rf_all>0.5; Average Revenue per Customer for New Model = mean(data_alldf$REVPERCUST[data_alldf$prediction_rf_all>0.5]), where the probability 0.5 is interchangeable and corresponds to the cutoffs above (blue text). Given the probability cutoffs mentioned above, XYZ should send the catalog to customers with a 5% and above predicted probability since this is the cutoff where profit is maximized using the random forest with SMOTE model. Doing so will result in $3,959,978 in profit. However, if XYZ wants to maximize average revenue per customer then they should use a cutoff of 65% instead and target that group accordingly.

Financial Evaluation Part II: XYZ Non-Targeted Population

	Random Forest Model Cutoff Rules							
Predicted Probability (percentage)	65.00	50.00	40.00	30.00	20.00	15.00	10.00	5.00
XYZ Current Targeting Methods								
Sample Size (All Customers Get Direct Mailing)	15857	15857	15857	15857	15857	15857	15857	15857
Average Revenue per Customer	118.8639	118.8639	118.8639	118.8639	118.8639	118.8639	118.8639	118.8639
Direct mail cost per Customer	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Ave. Revenue Minus Mail Cost per Customer	115.86	115.86	115.86	115.86	115.86	115.86	115.86	115.86
Profit with of Current Targeting Methods	$1,837,254	$1,837,254	$1,837,254	$1,837,254	$1,837,254	$1,837,254	$1,837,254	$1,837,254
XYZ Targeting with New Model								
Targeted Customers								
Number of Customers Targeted	1387	3118	5002	7931	11536	13335	14717	15598
Average Revenue per Customer	466.9753	339.6213	263.3613	191.4004	143.3182	129.8518	122.6703	119.5404
Direct mail cost per Customer	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Ave. Revenue Minus Mail Cost per Customer	463.98	336.62	260.36	188.40	140.32	126.85	119.67	116.54
Profit with New Model	$643,534	$1,049,585	$1,302,327	$1,494,204	$1,618,711	$1,691,569	$1,761,188	$1,817,797
Profit Increase or Loss with New Model	($1,193,720)	($787,669)	($534,927)	($343,050)	($218,543)	($145,685)	($76,066)	($19,457)
Per Customer Profit Contribution or Loss	($75.28)	($49.67)	($33.73)	($21.63)	($13.78)	($9.19)	($4.80)	($1.23)
Number of Customers in Database	30,779	30,779	30,779	30,779	30,779	30,779	30,779	30,779
Estimated Profit Contribution/Lift of Targeting	($2,317,053)	($1,528,893)	($1,038,312)	($665,873)	($424,200)	($282,780)	($147,647)	($37,766)

The table above shows financial evaluation/summary data for the random forest with SMOTE model so that we can identify which customers that XYZ *didn't* target in their most recent campaign so that XYZ can mail to those who will produce net revenue as a result. In order to obtain the results above, I first created another subset called subdat3 so that we can subset on the population that XYZ *didn't* target for campaign 16 (e.g., ANY_MAIL_16 == 0). We then applied the same data cleaning and imputation methods that we used for the targeted population set and applied the same random forest with SMOTE model from the targeted population to the population that XYZ didn't target in the 16th campaign. The model produced the following statistical evaluation criteria: AUC: 0.67757, AUPRC: 0.1790455, Sensitivity: 0.42535, and Specificity: 0.82129 on subdat3 dataset. Appendix 7 shows a histogram of probabilities for the random forest with SMOTE model (XYZ didn't target). Majority of the probabilities fall between 0.05 to 0.30 and then steadily decline. As a result, given the probability cutoffs above, XYZ should send the catalog to customers with a 5% and above predicted probability since this is the cutoff where profit is maximized. Doing so will result in $1,817,797 million in profit and a $19,457 loss with the new model. The profit is vastly lower than the targeted population, which makes sense given that both populations were most likely chosen based on cluster/segmentation analysis or possible selection bias. However, if XYZ wants to maximize average revenue per customer then they should use a cutoff of 65%. See Appendix 7 for ROC and Precision-Recall Curve as well.

Key Assumptions/Limitations: There are 4 key assumptions/limitations that we need to consider. First, similar to the targeted population, the non-targeted population suffers from an imbalanced dataset. For instance, it's important to note the prevalence is 0.07265, which indicates that only about 7% of customers bought items. However, given the fact that we utilized SMOTE on the random forest model, we were able to better predict the minority class. Second, it's important to keep in mind that we applied the same model from the targeted population (training) to the non-targeted population

(e.g., hold-out set). Therefore, we are making an assumption that a model that was trained on the targeted population, would also apply to the non-targeted population, but that may not always be the case. For instance, it's possible that both populations may differ. A third possible limitation is the fact that the two populations weren't chosen randomly. As a result, selection bias may be present. Lastly, a fourth possible limitation is that confounding effects and endogeneity may be present, which can affect the performance of our model in a negative way.

Financial Evaluation Part III

	Random Forest Model Cutoff Rules							
Predicted Probability (percentage)	65.00	50.00	40.00	30.00	20.00	15.00	10.00	5.00
XYZ Current Targeting Methods								
Sample Size (All Customers Get Direct Mailing)	14922	14922	14922	14922	14922	14922	14922	14922
Average Revenue per Customer	271.10	271.10	271.10	271.10	271.10	271.10	271.10	271.10
Direct mail cost per Customer	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Ave. Revenue Minus Mail Cost per Customer	268.10	268.10	268.10	268.10	268.10	268.10	268.10	268.10
Profit with of Current Targeting Methods	$4,000,657	$4,000,657	$4,000,657	$4,000,657	$4,000,657	$4,000,657	$4,000,657	$4,000,657
XYZ Targeting with New Model								
Targeted Customers								
Number of Customers Targeted	2332	4003	5353	7090	9416	10706	12120	13660
Average Revenue per Customer	699.7214	615.9711	537.7053	454.294	380.1277	350.9264	321.3439	292.8959
Direct mail cost per Customer	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Ave. Revenue Minus Mail Cost per Customer	696.72	612.97	534.71	451.29	377.13	347.93	318.34	289.90
Profit with New Model	$1,624,754	$2,453,723	$2,862,277	$3,199,674	$3,551,034	$3,724,900	$3,858,328	$3,959,978
Profit Increase or Loss with New Model	($2,375,903)	($1,546,934)	($1,138,379)	($800,982)	($449,622)	($275,757)	($142,329)	($40,679)
Per Customer Profit Contribution or Loss	($159.22)	($103.67)	($76.29)	($53.68)	($30.13)	($18.48)	($9.54)	($2.73)
Number of Customers in Database	30,779	30,779	30,779	30,779	30,779	30,779	30,779	30,779
Estimated Profit Contribution/Lift of Targeting	($4,900,677)	($3,190,797)	($2,348,089)	($1,652,154)	($927,418)	($568,792)	($293,576)	($83,907)

When comparing the net revenue from the current targeting method versus the random forest with SMOTE model for the targeted population (e.g., ANY_MAIL_16 > 0), it appears that the average revenue per customer using the model is much higher than the current targeting method at all probability cutoffs. On the other hand, XYZ's current targeting method produces more profit compared to the random forest model at all probability cutoffs. As a result, XYZ should mail to all customers since the net profit of doing so would be $4,000,657 million compared to $3,959,978 using the new model at a cutoff of 5%, which equates to a profit loss of $40,679 and per customer loss of $2.73. This was similarly seen in the non-targeted population as well (e.g., the current targeting method was better than the model at all probability cutoffs).

Interpretation of Most Preferred Model: Given that the most preferred model is the random forest with SMOTE model, the variable importance function can be used to determine the most important variables using MeanDecreaseAccuracy. MeanDecreaseAccuracy is a metric that shows how much each variable improves the prediction of its tree compared to the exact same tree without that variable. As a result, according to MeanDecreaseAccuracy (see Appendix 8): cum15QTY, PRE2009TRANSACTIONS, PRE2009SALES, and TOTAL_MAIL15 are considered the top 4 most important variables. This indicates that the sum of the number of items ordered for previous campaigns, total number of transactions before 2009, total sales before 2009, and total number of mailers received by the 15th campaign play a big role in determining whether or not a person will buy or not buy. This makes sense given that our EDA showed that people are more likely to buy as cum15QTY, PRE2009TRANSACTIONS, and PRESALES2009 increases. From a marketing standpoint this makes sense, given that past spending behavior is indicative of future behavior. In other words, the more someone spent in the past, the more likely they are to buy in the future. Additionally, the more someone receives mailings in the previous campaigns, the more they are likely to spend since they are constantly exposed to the company's products. Lastly, the fact that MED_HOME and INC_WITHOUT_V4 had high MeanDecreaseAccuracy scores and appeared in both stepAIC (logistic regression) and varimp's top 20 (random forest) illustrates that both variables are also strong predictors. This makes sense given that buyers must be able to afford the electronic appliances that XYZ is selling. For instance, we saw in our EDA that people who fall in different groups within INC_WIOUTSCS_V4 are more willing to buy compared to others. Additionally, as median housing value goes up, people are more likely to buy. Overall, XYZ should utilize these insights to alter their advertising and targeting strategy.

Target Marketing Recommendation & Summary: Overall, if XYZ wants to maximize profit they should use the current targeting method and send the catalog to all customers. However, if XYZ wants to maximize average revenue per customer, they should target customers that have a probability of 65% or higher. I also noticed that revenue per customer increases as the probability increases (e.g., 70%, 80%, 90%, etc.), but the number of customers targeted decreases. As a result, my recommendation to XYZ would be to send "special catalogs" tailored to the high revenue generating buyers (65-85% probability) and invite their highest probability customers (85% probability and above) to a special event. For example, Nordstrom invites their most loyal and highest spending customers to an "exclusive" shopping event at the end of the year, where majority of them end up buying merchandise. Therefore, XYZ should target customers that spent/bought a lot of merchandise in the past since people are more likely to buy as cum15QTY, PRE2009TRANSACTIONS, and PRESALES2009 increases. Lastly, XYZ, should target customers with high incomes and live in areas with high median

housing value. For instance, we saw in our EDA that people who fall in different groups within INC_WIOUTSCS_V4 are more willing to buy compared to others and as median housing value goes up, people are more likely to buy.

Applying Model to New Customers & Future Work: In order for XYZ to apply the final model (random forest with SMOTE) to new customers they must first collect the necessary customer data. For instance, all the variables that are included in the random forest with SMOTE model must be obtained. Next, we must clean and impute the new customer data using similar methods that we used for our final model. Upon having a "clean" new customer dataset, the random forest with SMOTE model can then be applied. On a side note, given that we were having difficulties predicting the "rare events", it could be helpful to obtain more data (especially those who bought items) and experiment with other resampling techniques (e.g., random under-sampling, random over-sampling, cluster-based over sampling) or MSMOTE: Modified synthetic minority oversampling technique) prior to rolling out the final model to new customers. It could also be helpful to enlist a marketing consultant or a team of marketing experts to help properly choose the correct variables out of the 554 variables so that the model could be improved. Lastly, it could be helpful experimenting with other ensemble techniques (e.g., bagging, boosting, XG Boost, etc.), try penalized models (penalized SVM, LDA, etc.), or try a combination of resampling + ensemble methods.

Assessing Models Ability to Predict Responses to XYZ's Next (17th Campaign): My recommendation to XYZ in order to assess the models ability to predict responses to XYZ's next (17th campaign) would be to design an experiment or "test" between two groups. The control group may be a random sample of potential customers from the marketing campaign list that is excluded from the mailing list for the 17th campaign. The control group's results can then be compared to the experimental group (e.g., group that was targeted for the 17th campaign). I would also advise conducting propensity score matching to reduce selection bias given that the two groups may be different, but there may be characteristics that are similar. For instance, propensity scoring matches the appropriate variables so that we could ensure that two people have an equal chance of being in the control or experimental group and helps us determine whether the results that we are receiving are due to one group being more inclined due to the fact they are a part of the control or experimental group versus the effect of the outcome (buyers vs. non-buyers) itself. This will allow the marketing team to accurately determine how effective and profitable the campaign is. We could also compare statistical (lift charts, ROC Curves, AUC, AUPRC, Accuracy, Log loss) and financial evaluation criteria (number of customers targeted, average revenue per customer, net revenue generated for the given probability cutoffs) between the two groups, current targeting method, and over time (monthly).

Conclusion

In conclusion, we began this analysis by first conducting EDA to get a better feel of the dataset. We then performed feature engineering, feature selection, data cleaning (e.g., missing value imputation). We then created and validated 3 classification models (logistic regression, LDA, and random forest with SMOTE). The results, showed that random forest with SMOTE performed the best based on sensitivity (true positive rate). We then determined that using a probability cutoff of 5% would generate the most profit for the targeted and non-targeted population in the 16th campaign. We then compared the expected net revenue from the most preferred model versus the actual revenue that resulted from XYZ's 16th campaign and determined that using the current targeting method of mailing to all customers would result in higher net profit. This was similarly seen in the non-targeted population as well (e.g., the current targeting method was better than the model at all probability cutoffs). We then interpreted the random forest with SMOTE model results, made recommendations on how to apply the model to new customers, potential future work, and how to assess the models ability to predict responses to XYZ's next (17th campaign).

References

1. Brownlee, J. (2015, August 19). *8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset*. Retrieved from https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/.

2. Fawcett, T. (2005). *An Introduction to ROC analysis*. Retrieved from https://people.inf.elte.hu/kiss/11dwhdm/roc.pdf.

3. Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. New York: Springer Science + Business Media.

4. James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. New York: Springer Science + Business Media.

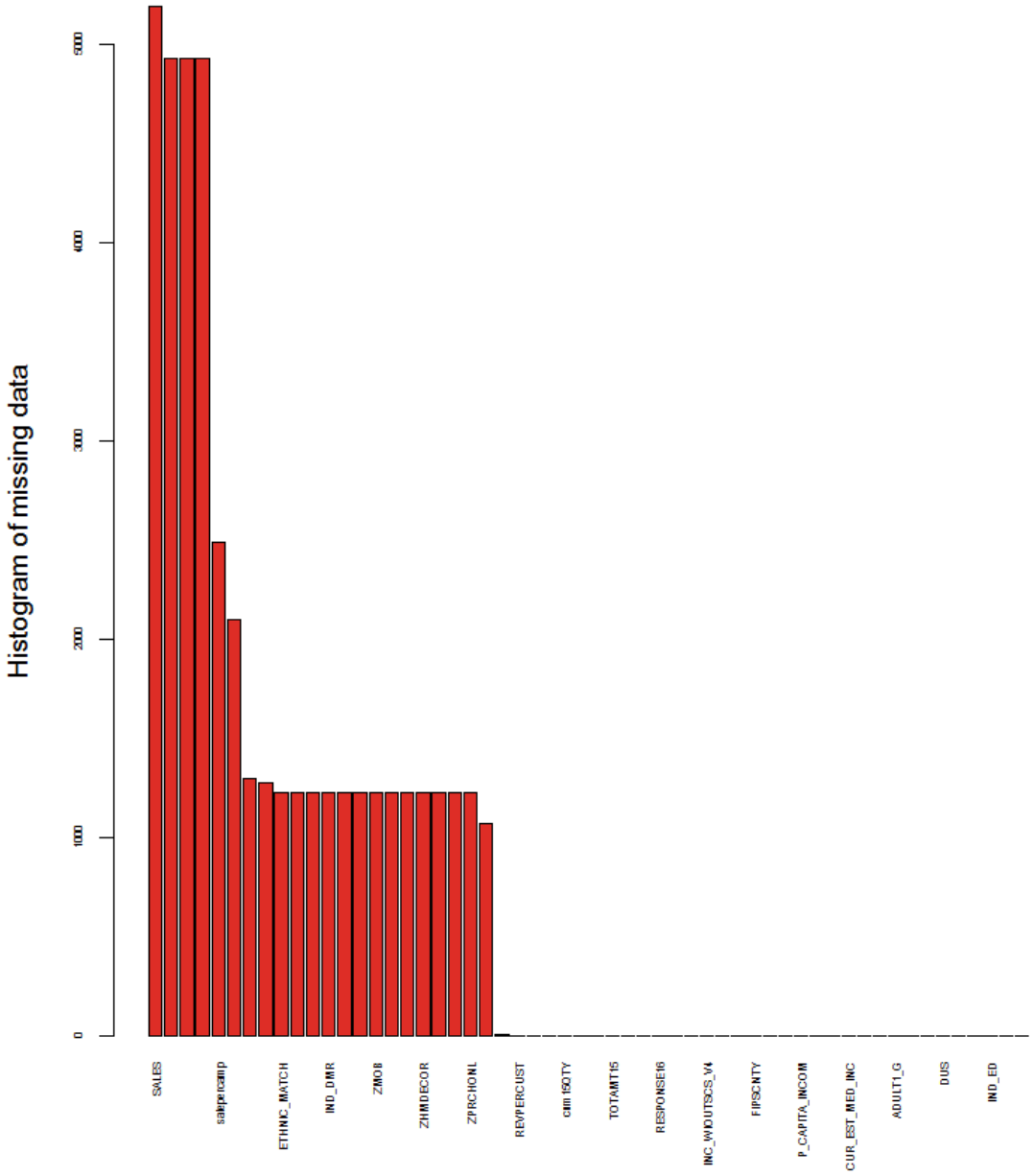
5. Mukherjee, U. (2017, March 17). *How to handle Imbalanced Classification Problems in machine learning?*. Retrieved from https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/.

6. Saito, T., Rehmsmeier, M. (2015). *The Precision-Recall Plot Is More Informative than the ROC Plot When E valuating Binary Classifiers on Imbalanced Datasets*. Retrieved from https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0118432.

Appendix

Appendix 1

The histogram in Appendix 1 shows the total number of missing values for all the variables in the dataset using the VIM package. For instance, the following variables had missing values: SALES (5191), DEBIT_CC (4931), MAJOR_CC (4931), COMPUTER_ELECTRONIC (4931), salepercamp (2487), EXAGE (2097), salepertrans (1294), CHANNEL_DOMINANCE (1274), *ETHNIC_MATCH (1225), HOMEOWNR, ADD_TYPE, IND_DMR, MAILPREF, ZONLINE, ZMOB, ZPRCHPHN, ZMOBMULT, ZHMDECOR, ZHOMEENT, ZKITCHEN, ZPRCHONL*, NUM_CHILD (1071), and ZIP (1). *Note: Italicized variables each had 1225 missing values.*



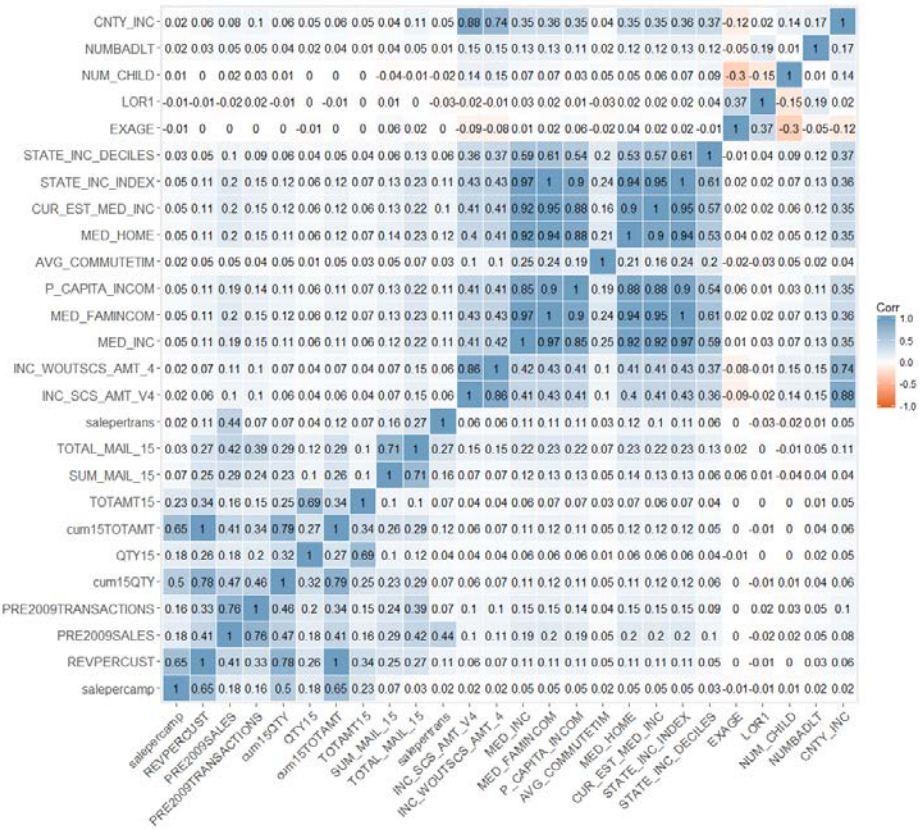
Appendix 2

Appendix 2, shows summary statistics of the numeric variables in the training dataset so that we can check for missing values, outliers, distributions, etc. For instance, the data shows that the mean for salespercamp is 37.83, salepertrans is 251.7, REVPERCUST is 265.8, PRE2009SALES is 1276, PRE2009TRANSACTIONS is 4.492, MED_HOME is 345792, cum15QTY is 2.517, and INC_WOUTSCS_AMT_4 is 140.6.

```
  salepercamp        REVPERCUST       PRE2009SALES   PRE2009TRANSACTIONS     cum15QTY          QTY15          cum15TOTAMT
Min.   :   0.00   Min.   : -90.0   Min.   :    0   Min.   :  0.000      Min.   :  0.000   Min.   : 0.000   Min.   :    0.0
1st Qu.:   0.00   1st Qu.:  -9.0   1st Qu.:  183   1st Qu.:  1.000      1st Qu.:  0.000   1st Qu.: 0.000   1st Qu.:    0.0
Median :   0.00   Median :   0.0   Median :  531   Median :  3.000      Median :  0.000   Median : 0.000   Median :    0.0
Mean   :  37.83   Mean   : 265.8   Mean   : 1276   Mean   :  4.492      Mean   :  2.517   Mean   : 0.172   Mean   :  258.0
3rd Qu.:  34.00   3rd Qu.: 248.4   3rd Qu.: 1380   3rd Qu.:  6.000      3rd Qu.:  3.000   3rd Qu.: 0.000   3rd Qu.:  243.9
Max.   :4263.27   Max.   :16261.2   Max.   :39402   Max.   :178.000      Max.   :117.000   Max.   :32.000   Max.   :14815.6
    TOTAMT15        SUM_MAIL_15      TOTAL_MAIL_15     salepertrans     INC_SCS_AMT_V4  INC_WOUTSCS_AMT_4     MED_INC        MED_FAMINCOM
Min.   :   0.00   Min.   :0.0000   Min.   :    0.000   Min.   :   0.0   Min.   :  0.0   Min.   :  0.0     Min.   : 13749   Min.   : 18853
1st Qu.:   0.00   1st Qu.:0.0000   1st Qu.:    1.000   1st Qu.:  99.0   1st Qu.: 86.0   1st Qu.: 86.0     1st Qu.: 64862   1st Qu.: 75687
Median :   0.00   Median :0.0000   Median :    4.000   Median : 181.5   Median :125.0   Median :125.0     Median : 86939   Median : 99068
Mean   :  19.71   Mean   :0.3105   Mean   :    6.018   Mean   : 251.7   Mean   :140.1   Mean   :140.6     Mean   : 97650   Mean   :111695
3rd Qu.:   0.00   3rd Qu.:1.0000   3rd Qu.:   11.000   3rd Qu.: 305.7   3rd Qu.:195.0   3rd Qu.:195.0     3rd Qu.:115548   3rd Qu.:133377
Max.   :6383.37   Max.   :3.0000   Max.   :   45.000   Max.   :10232.0   Max.   :250.0   Max.   :250.0     Max.   :244999   Max.   :274166
P_CAPITA_INCOM   AVG_COMMUTETIM     MED_HOME      CUR_EST_MED_INC  STATE_INC_INDEX STATE_INC_DECILES      EXAGE            LOR1
Min.   :  1272   Min.   : 50.0   Min.   :     0   Min.   : 16666   Min.   : 33.0   Min.   :0.000     Min.   :19.00   Min.   : 0.00
1st Qu.: 30289   1st Qu.:252.0   1st Qu.:210549   1st Qu.: 91428   1st Qu.:133.0   1st Qu.:7.000     1st Qu.:46.00   1st Qu.: 6.00
Median : 37804   Median :282.0   Median :281249   Median :120398   Median :174.0   Median :9.000     Median :53.00   Median :12.00
Mean   : 45204   Mean   :281.4   Mean   :345792   Mean   :136824   Mean   :196.8   Mean   :8.044     Mean   :53.27   Mean   :13.71
3rd Qu.: 50798   3rd Qu.:305.0   3rd Qu.:425788   3rd Qu.:166775   3rd Qu.:235.0   3rd Qu.:9.000     3rd Qu.:61.00   3rd Qu.:19.00
Max.   :127968   Max.   :449.0   Max.   :999999   Max.   :437499   Max.   :484.0   Max.   :9.000     Max.   :98.00   Max.   :54.00
   NUM_CHILD        NUMBADLT        CNTY_INC
Min.   :0.0000   Min.   :0.0   Min.   :  0.00
1st Qu.:0.0000   1st Qu.:2.0   1st Qu.: 62.00
Median :0.0000   Median :3.0   Median : 84.00
Mean   :0.5085   Mean   :2.9   Mean   : 74.58
3rd Qu.:1.0000   3rd Qu.:4.0   3rd Qu.: 93.00
Max.   :7.0000   Max.   :8.0   Max.   : 99.00
```

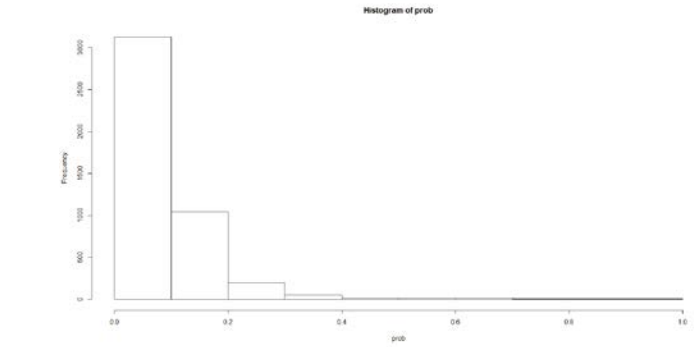
Appendix 3

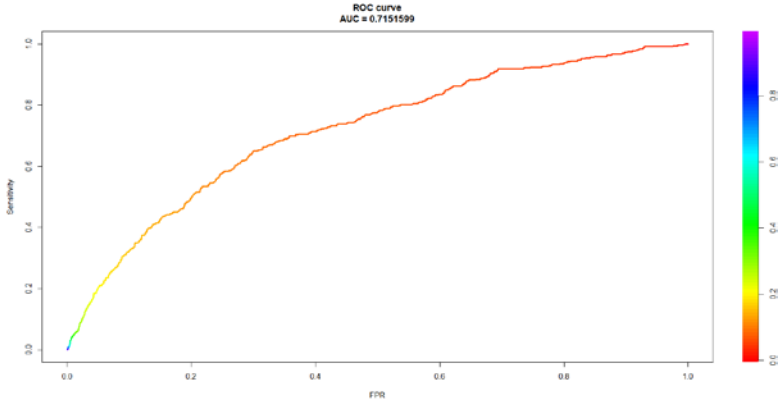
The correlation matrix in Appendix 3 allows us to see which variables may be correlated with each other so that I can gleam interesting insights. The plot shows that the sales versus transaction history variables have strong positive correlations with each other, while many of the income related variables have strong positive correlations with each other as well. *Blue = positive correlations and red = negative correlations. See key below.*

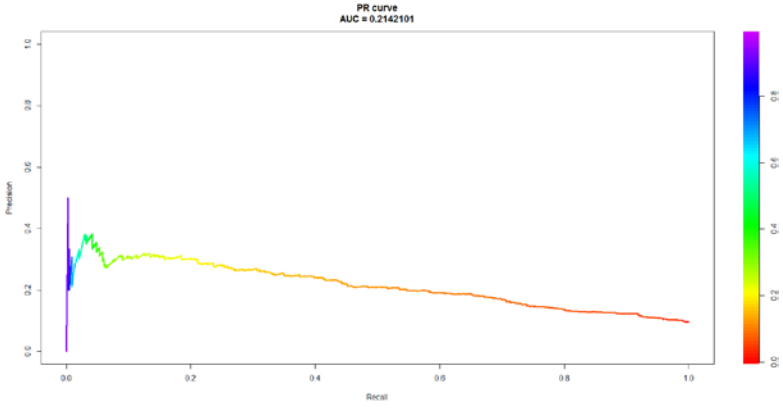


Appendix 4

Appendix 4 shows a histogram of probabilities for the logistic regression model on the validation set. Majority of the probabilities fall in-between 0 to 0.2. See below for ROC and Precision-Recall Curve as well.

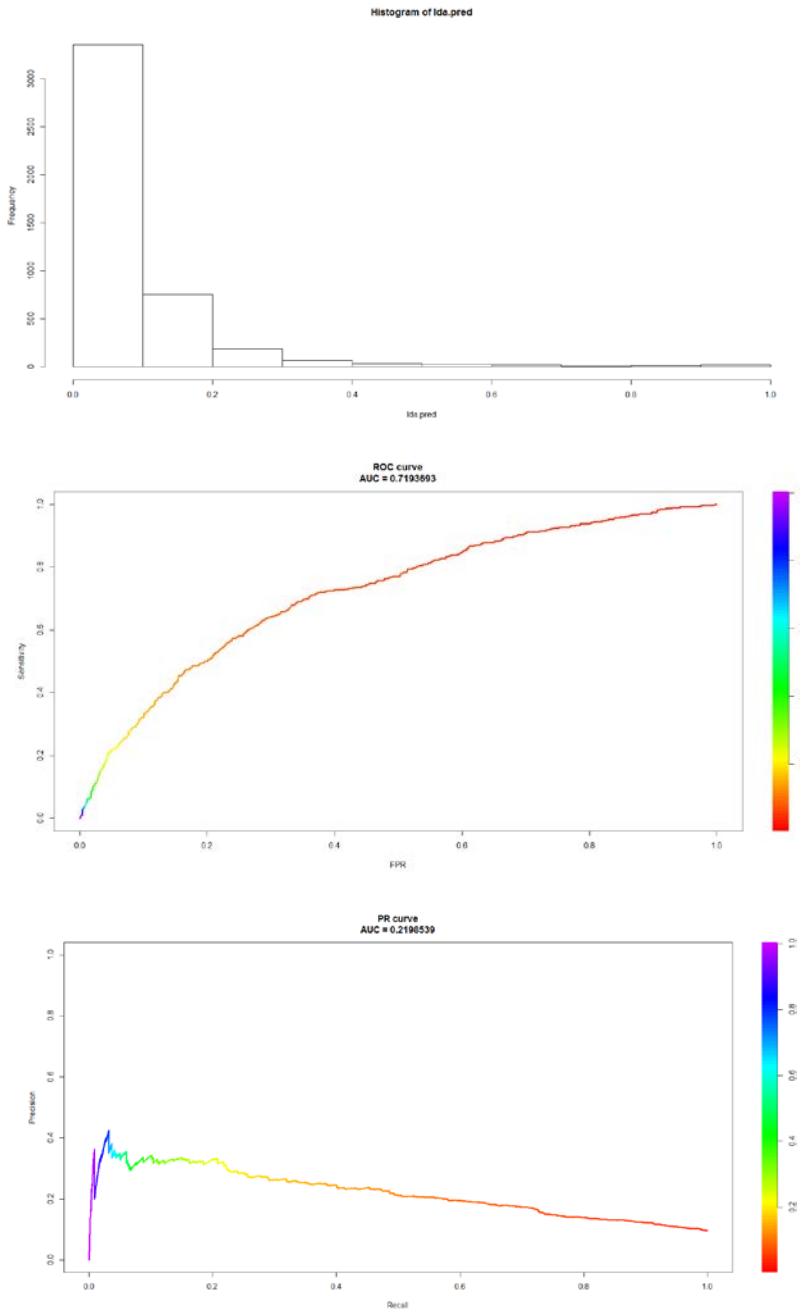






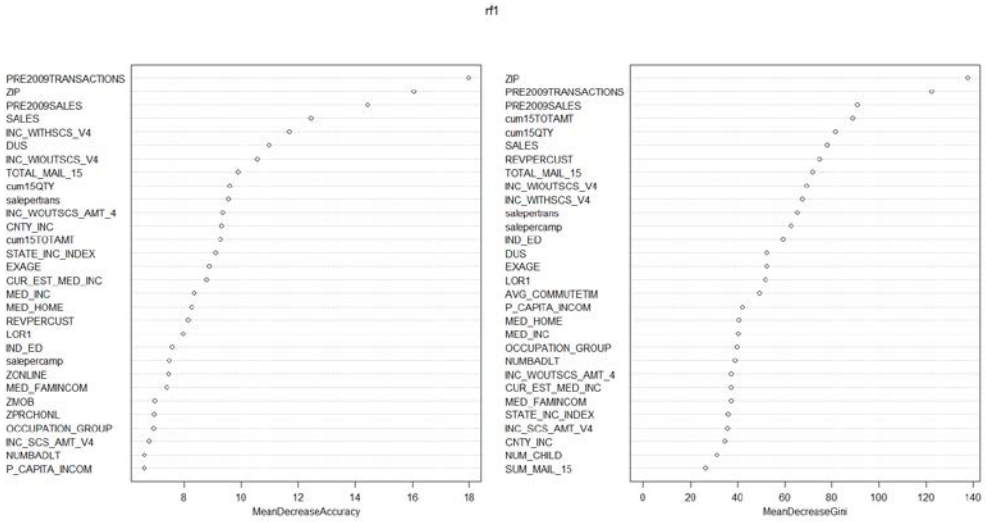
Appendix 5

Appendix 5 shows a histogram of probabilities for the LDA model on the validation set. Majority of the probabilities fall in-between 0 to 0.2. See below for ROC and Precision-Recall Curve as well.

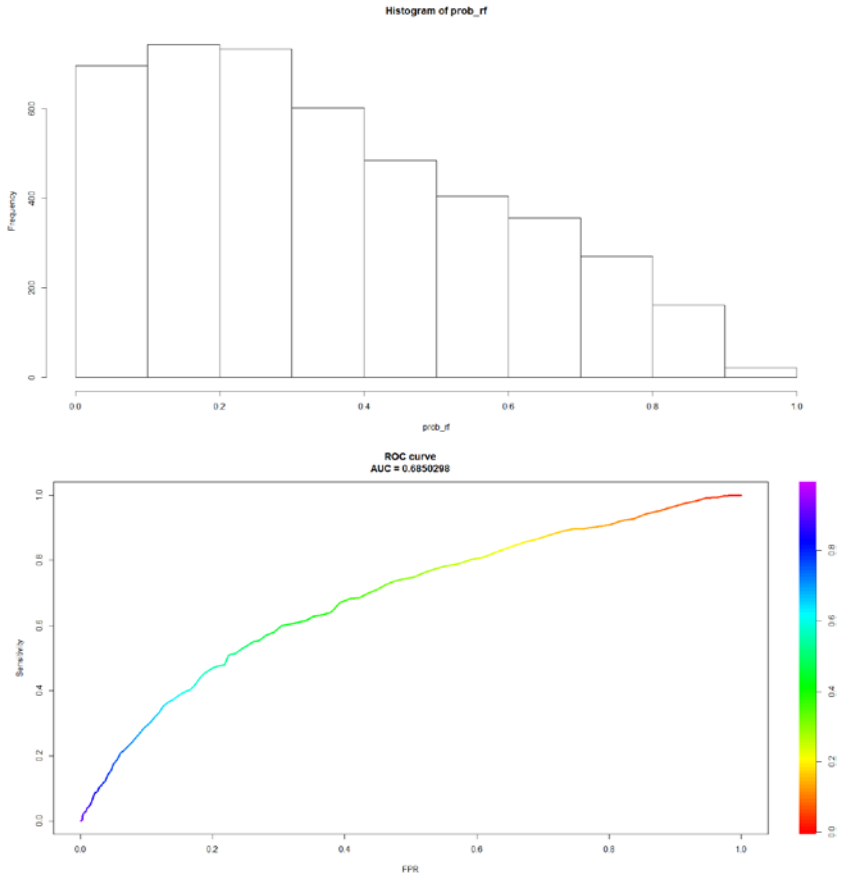


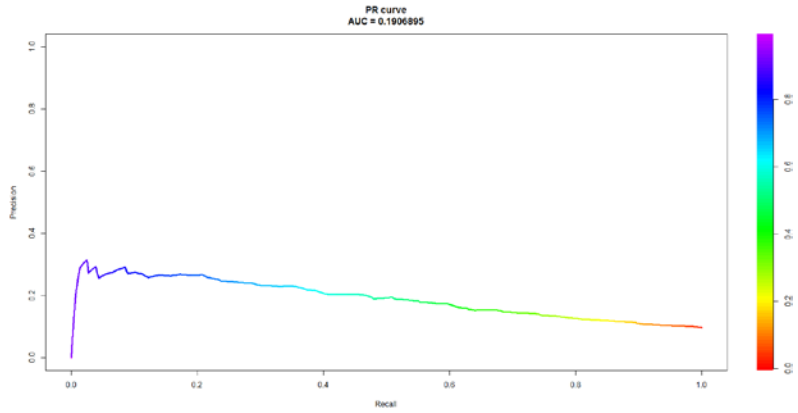
Appendix 6

MeanDecreaseAccuracy is a metric that shows how much each variable improves the prediction of its tree compared to the exact same tree without that variable. According to the variable importance function (which measures prediction strength), the plot below shows the top 30 most important predictors using random forest (varimp function).



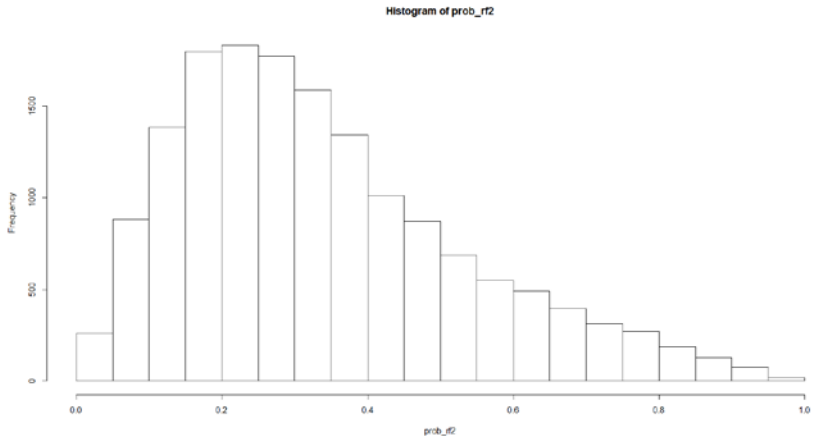
The histogram below shows the probabilities for the random forest with SMOTE model on the validation set. Majority of the probabilities fall near 0 to 0.3 and then steadily decline. See below for ROC and Precision-Recall Curve as well.

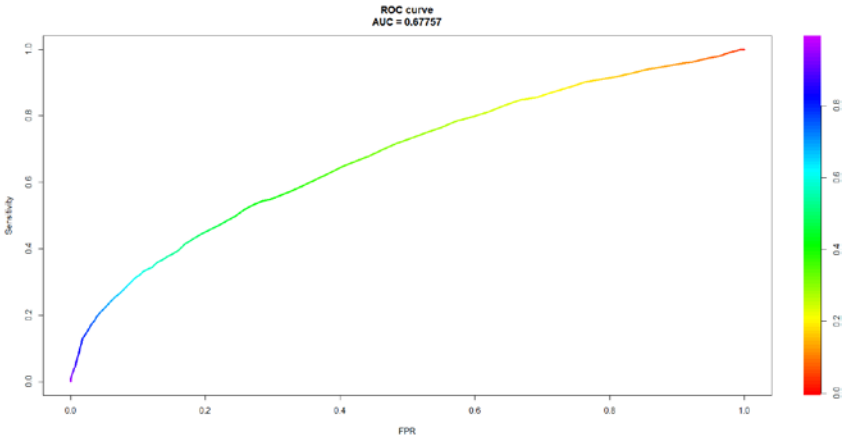


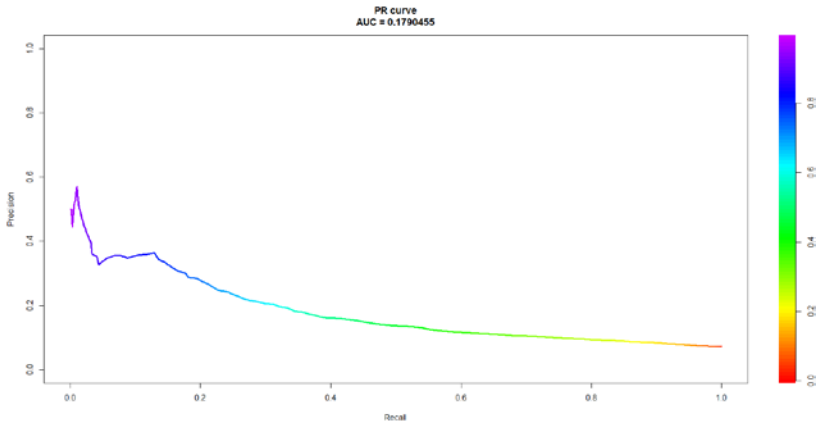


Appendix 7

Appendix 7 shows a histogram of probabilities for the random forest with SMOTE model (XYZ didn't target). Majority of the probabilities fall between 0.05 to 0.30 and then steadily decline. See below for ROC and Precision-Recall Curve as well.

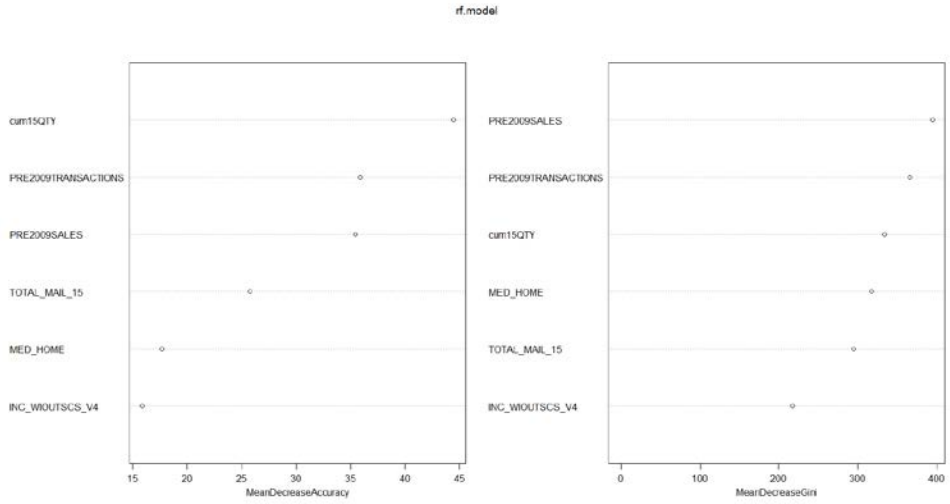






Appendix 8

MeanDecreaseAccuracy is a metric that shows how much each variable improves the prediction of its tree compared to the exact same tree without that variable. As a result, according to MeanDecreaseAccuracy, cum15QTY, PRE2009TRANSACTIONS, PRE2009SALES, and TOTAL_MAIL15 are considered the top 4 most important variables for the random forest with SMOTE model.



R Code

```
#Brent Young
#MSDS 450, Winter 2019
#Solo 3 Assignment

library(Hmisc)
library(VIM) #Missingness Map
library (ggplot2) #Data Visualization

# Multiple plot function

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}


###### Load Data #####

setwd("~/R/MSDS 450/Solo 3")
load("XYZ_complete_customer_data_frame.RData")
ls()
mydata <- complete.customer.data.frame #30779 observations and 554 variables
names(mydata) #column names

#write.csv(mydata, file="BDY.csv") # use your initials for the file name
```

```
###################################### Exploratory Data Analysis ################
####################

##### High Level Descriptive Statistics #####
str(mydata)
summary(mydata)
#describe(mydata)
dim(mydata) #30779 customers, 554 variables

#Table by data type
type_table <- sapply(mydata, typeof)
table(type_table)
#type_table
#character    double   integer
#345       161        48

#Buyer Status
table(mydata$BUYER_STATUS) #Active, Inactive, and Lapsed

#16th Campaign - Response Variable (1= Responded, 0 = Did Not Respond)
table(mydata$RESPONSE16)
#    0     1
#28187  2592

#Customers Targeted in 16th Campaign (1=Received Mailer, 0=Did Not Receive Mailer)
table(mydata$ANY_MAIL_16)
#    0     1
#15857 14922

#16th Campaign - Response vs. Targeted
xtabs(~RESPONSE16 + ANY_MAIL_16, data = mydata)
#14705 people did not respond and weren't targeted
#13483 people did not respond and were targeted
#1152 people did respond and weren't targeted
#1440 people did respond and were targeted

xtabs(~RESPONSE16 + ANY_MAIL_16, data = mydata)

##### Feature Engineering #####

#cum15TOTAMT
#Sum of Total Sales $ By Customer for Previous Campaigns
mydata$cum15TOTAMT <- mydata$TOTAMT0 + mydata$TOTAMT1 + mydata$TOTAMT2 + mydata$TO
TAMT3 +
  mydata$TOTAMT4 + mydata$TOTAMT5 + mydata$TOTAMT6 + mydata$TOTAMT7 + mydata$TOTAM
T8 +
  mydata$TOTAMT9 + mydata$TOTAMT10 + mydata$TOTAMT11 + mydata$TOTAMT12 + mydata$TO
TAMT13 +
  mydata$TOTAMT14 + mydata$TOTAMT15

#REVPERCUST
mydata$REVPERCUST = mydata$cum15TOTAMT*1.10 - (mydata$TOTAL_MAIL_15*3.00)

#PRE2009SALES
#Total Sales Before 2009
mydata$PRE2009SALES = mydata$LTD_SALES - mydata$YTD_SALES_2009

#PRE2009TRANSACTIONS
#Total # of Transactions Before 2009
```

```
mydata$PRE2009TRANSACTIONS = mydata$LTD_TRANSACTIONS -
  mydata$YTD_TRANSACTIONS_2009

#cum15QTY
#Sum of # of Items Ordered for Previous Campaigns
mydata$cum15QTY <- mydata$QTY0 + mydata$QTY1 + mydata$QTY2 + mydata$QTY3 +
  mydata$QTY4 + mydata$QTY5 + mydata$QTY6 + mydata$QTY7 + mydata$QTY8 +
  mydata$QTY9 + mydata$QTY10 + mydata$QTY11 + mydata$QTY12 + mydata$QTY13 +
  mydata$QTY14 + mydata$QTY15

#salepertrans
#Sales Per Transaction
mydata$salepertrans <- mydata$PRE2009SALES/mydata$PRE2009TRANSACTIONS

#salepercamp
#Sum of Total Sales for 15th Campaign
mydata$salepercamp <- mydata$cum15TOTAMT/mydata$TOTAL_MAIL_15

#Descriptive Statistics of Created Variables
meansalepercust <- mean(mydata$PRE2009SALES) #Average Dollars Spent Per Customer #
979.2211
meansalepercust
minsalepercust <- min(mydata$PRE2009SALES) #0
minsalepercust
maxsalepercust <- max(mydata$PRE2009SALES) # 94350
maxsalepercust

mean(mydata$REVPERCUST) #Average Revenue Per Customer for Previous Campaigns #192.
6719
mean(mydata$PRE2009TRANSACTIONS) #Average Transactions Per Customer for Previous C
ampaigns #3.800546
mean(mydata$cum15TOTAMT) #Average Total Sales Per Customer for Previous Campaignsi
s #187.1452
mean(mydata$cum15QTY) #Average Quantity Purchased for Previous Campaigns #1.941389

#Trimmed and Remove Outliers and NA's
mean(mydata$salepertrans, trim = 0.01, na.rm = TRUE) #Average of sale of $233 per
transaction
mean(mydata$salepercamp, trim = 0.05, na.rm = TRUE) #Average sales for cumulative
15th campaign #$36

#Load Libraries
require(rpart)
require(rpart.plot)
require(tree)
require(rattle)
require(caTools)
require(ROCR)
require(ResourceSelection) #hoslem.test
library(PRROC) #ROC and Precision Recall Curve

library(corrgram)
library(MASS)
library(randomForest)
library(inTrees)
library(pROC)
library(caret)
library(dplyr)
```

```
################################################################################
#############
###Step 1 - Pretend you are a domain expert and pick upto 50 'important' predictor
s
################################################################################
#################

subdat <- subset(mydata, select=c("ZIP","REVPERCUST", "salepercamp", "PRE2009SALES
","PRE2009TRANSACTIONS","cum15QTY","QTY15",
                                  "cum15TOTAMT","TOTAMT15","SUM_MAIL_15","TOTAL_MA
IL_15","RESPONSE16","salepertrans",
                                  "CHANNEL_ACQUISITION","ANY_MAIL_16",
                                  "DEBIT_CC","MAJOR_CC","COMPUTER_ELECTRONIC","INC
_SCS_AMT_V4","INC_WIOUTSCS_V4",
                                  "INC_WITHSCS_V4","INC_WOUTSCS_AMT_4","FIPSCNTY",
"MED_INC","MED_FAMINCOM","P_CAPITA_INCOM",
                                  "AVG_COMMUTETIM","MED_HOME","CUR_EST_MED_INC","S
TATE_INC_INDEX","STATE_INC_DECILES",
                                  "EXAGE","ADULT1_G","MARRIED","ETHNIC_MATCH","HOM
EOWNR","ADD_TYPE",
                                  "LOR1","DUS","NUM_CHILD","NUMBADLT","IND_DMR","O
CCUPATION_GROUP",
                                  "IND_ED","PRESCHLD","CNTY_INC","MAILPREF","CHANN
EL_DOMINANCE","SALES",
                                  "ZONLINE","ZMOB","ZPRCHPHN","ZMOBMULT","ZHMDECOR
","ZHOMEENT",
                                  "ZKITCHEN","ZPRCHONL"))


str(subdat)

################################################################################
#############
############### Step 2 - Pick only the people who were sent mailers in Campaign 16
##########
################################################################################
#############

subdat2 <- subset(subdat, ANY_MAIL_16 > 0)
str(subdat2)

##### Change Variable Types #####
#Factors
subdat2$ZIP <- as.factor(subdat2$ZIP)
subdat2$RESPONSE16 <- as.factor(subdat2$RESPONSE16)
subdat2$CHANNEL_ACQUISITION <- as.factor(subdat2$CHANNEL_ACQUISITION)
subdat2$DEBIT_CC <- as.factor(subdat2$DEBIT_CC)
subdat2$MAJOR_CC <- as.factor(subdat2$MAJOR_CC)
subdat2$COMPUTER_ELECTRONIC <- as.factor(subdat2$COMPUTER_ELECTRONIC)
subdat2$INC_WIOUTSCS_V4 <- as.factor(subdat2$INC_WIOUTSCS_V4)
subdat2$INC_WITHSCS_V4 <- as.factor(subdat2$INC_WITHSCS_V4)
subdat2$FIPSCNTY <- as.factor(subdat2$FIPSCNTY)
subdat2$ADULT1_G <- as.factor(subdat2$ADULT1_G)
subdat2$MARRIED <- as.factor(subdat2$MARRIED)
subdat2$ETHNIC_MATCH <- as.factor(subdat2$ETHNIC_MATCH)
subdat2$HOMEOWNR <- as.factor(subdat2$HOMEOWNR)
subdat2$ADD_TYPE <- as.factor(subdat2$ADD_TYPE)
subdat2$DUS <- as.factor(subdat2$DUS)
subdat2$IND_DMR <- as.factor(subdat2$IND_DMR)
```

```
subdat2$OCCUPATION_GROUP <- as.factor(subdat2$OCCUPATION_GROUP)
subdat2$IND_ED <- as.factor(subdat2$IND_ED)
subdat2$PRESCHLD <- as.factor(subdat2$PRESCHLD)
subdat2$MAILPREF <- as.factor(subdat2$MAILPREF)
subdat2$CHANNEL_DOMINANCE <- as.factor(subdat2$CHANNEL_DOMINANCE)
subdat2$SALES <- as.factor(subdat2$SALES)
subdat2$ZONLINE <- as.factor(subdat2$ZONLINE)
subdat2$ZMOB <- as.factor(subdat2$ZMOB)
subdat2$ZPRCHPHN <- as.factor(subdat2$ZPRCHPHN)
subdat2$ZMOBMULT <- as.factor(subdat2$ZMOBMULT)
subdat2$ZHMDECOR <- as.factor(subdat2$ZHMDECOR)
subdat2$ZHOMEENT <- as.factor(subdat2$ZHOMEENT)
subdat2$ZKITCHEN <- as.factor(subdat2$ZKITCHEN)
subdat2$ZPRCHONL <- as.factor(subdat2$ZPRCHONL)

#Numeric
subdat2$EXAGE <- as.numeric(subdat2$EXAGE)
subdat2$LOR1 <- as.numeric(subdat2$LOR1)
subdat2$NUM_CHILD <- as.numeric(subdat2$NUM_CHILD)
subdat2$SUM_MAIL_15 <- as.numeric(subdat2$SUM_MAIL_15)
subdat2$TOTAL_MAIL_15 <- as.numeric(subdat2$TOTAL_MAIL_15)
subdat2$REVPERCUST <- as.numeric(subdat2$REVPERCUST)
subdat2$salepercamp <- as.numeric(subdat2$salepercamp)

str(subdat2)
#Remove Variables
subdat2$ANY_MAIL_16 <- NULL

#subdat2 <- subset(subdat, EXAGE > 0)
##unitamtt <- mean(subdat2$TOTAMT)

#write.csv(subdat2, file="BDY2.csv") # use your initials for the file name

################################################################################
#################### Step 3 - EDA - Cleaning up of the data  ####################
################################################################################

str(subdat2)

# Convert Blanks to NA
subdat2[subdat2 == ""] <- NA

#Check for Missingness
sum(is.na(subdat2))
sapply(subdat2, function(x) sum(is.na(x)))
aggr_plot <- aggr(subdat2, col=c('#9ecae1','#de2d26'), numbers=TRUE,prop=FALSE, so
rtVars=TRUE, labels=names(subdat2), cex.axis=.5, gap=2, ylab=c("Histogram of missi
ng data","Pattern"))

#Check missing data percentage
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(subdat2,2,pMiss)

#Impute Factor Variables with Majority Class (except ETHNIC_MATCH)
subdat2$DEBIT_CC[is.na(subdat2$DEBIT_CC)] <- "U" #Majority
subdat2$MAJOR_CC[is.na(subdat2$MAJOR_CC)] <- "U" #Majority
subdat2$COMPUTER_ELECTRONIC[is.na(subdat2$COMPUTER_ELECTRONIC)] <- "U" #Majority
subdat2$CHANNEL_DOMINANCE[is.na(subdat2$CHANNEL_DOMINANCE)] <- "C" #Majority
```

```
levels(subdat2$ETHNIC_MATCH)<-c(levels(subdat2$ETHNIC_MATCH),"N")
subdat2$ETHNIC_MATCH[is.na(subdat2$ETHNIC_MATCH)] <- "N" #Blanks = No

subdat2$HOMEOWNR[is.na(subdat2$HOMEOWNR)] <- "Y" #Majority
subdat2$ADD_TYPE[is.na(subdat2$ADD_TYPE)] <- "S" #Majority
subdat2$IND_DMR[is.na(subdat2$IND_DMR)] <- "U" #Majority
subdat2$MAILPREF[is.na(subdat2$MAILPREF)] <- "N" #Majority

subdat2$ZONLINE[is.na(subdat2$ZONLINE)] <- "Y" #Majority
subdat2$ZMOB[is.na(subdat2$ZMOB)] <- "Y" #Majority
subdat2$ZPRCHPHN[is.na(subdat2$ZPRCHPHN)] <- "U" #Majority
subdat2$ZMOBMULT[is.na(subdat2$ZMOBMULT)] <- "Y" #Majority
subdat2$ZHMDECOR[is.na(subdat2$ZHMDECOR)] <- "Y" #Majority
subdat2$ZHOMEENT[is.na(subdat2$ZHOMEENT)] <- "U" #Majority
subdat2$ZKITCHEN[is.na(subdat2$ZKITCHEN)] <- "U" #Majority
subdat2$ZPRCHONL[is.na(subdat2$ZPRCHONL)] <- "Y" #Majority
subdat2$SALES[is.na(subdat2$SALES)] <- "U" #Unknown

subdat2$ZIP[is.na(subdat2$ZIP)] <- "60091" #Majority

#Impute Numeric Variables with Median
subdat2$EXAGE[subdat2$EXAGE=="U"] <- NA
subdat2$EXAGE[is.na(subdat2$EXAGE)] = median(subdat2$EXAGE, na.rm = TRUE)

#Impute Numeric Variables with 0
subdat2$salepercamp[is.na(subdat2$salepercamp)] <- 0
subdat2$salepercamp[is.infinite(subdat2$salepercamp)] <- 0
subdat2$salepertrans[is.na(subdat2$salepertrans)] <- 0
subdat2$NUM_CHILD[is.na(subdat2$NUM_CHILD)] <- 0

#Check for Missingness
sum(is.na(subdat2))
sapply(subdat2, function(x) sum(is.na(x)))
aggr_plot <- aggr(subdat2, col=c('#9ecae1','#de2d26'), numbers=TRUE,prop=FALSE, so
rtVars=TRUE, labels=names(subdat2), cex.axis=.5, gap=2, ylab=c("Histogram of missi
ng data","Pattern"))

#Check missing data percentage
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(subdat2,2,pMiss)

str(subdat2)

#subdat2$logPRE2009TRANSACTIONS <- log(subdat2$PRE2009TRANSACTIONS) #Possible log
transformation

######################################## Split Datasets ##########################
############
#Split Data into Train/Test
#train <- subset(subdat2, LEARNING_TEST=="LEARNING") #splits data into training da
taset
#test <- subset(subdat2, LEARNING_TEST=="TEST") #splits data into test dataset

set.seed(123)
validationIndex <- createDataPartition(subdat2$RESPONSE16, p=0.70, list=FALSE)
train <- subdat2[validationIndex,]
validation <- subdat2[-validationIndex,]

str(train)
```

```
str(validation)

sum(is.na(train))
sum(is.na(validation))

summary(train)

############################## EDA for Classification Models - Training Data #####
#########################

##################### EDA Numeric ################################

subdatnumcor <- subset(train, select=c("salepercamp","REVPERCUST","PRE2009SALES","
PRE2009TRANSACTIONS","cum15QTY", "QTY15",
                                       "cum15TOTAMT", "TOTAMT15","SUM_MAIL_15","TO
TAL_MAIL_15",
                                       "salepertrans","INC_SCS_AMT_V4","INC_WOUTSC
S_AMT_4","MED_INC","MED_FAMINCOM",
                                       "P_CAPITA_INCOM","AVG_COMMUTETIM","MED_HOME
","CUR_EST_MED_INC","STATE_INC_INDEX","STATE_INC_DECILES",
                                       "EXAGE","LOR1","NUM_CHILD","NUMBADLT","CNTY
_INC"))

str(subdatnumcor)
summary(subdatnumcor)

par(mfrow=c(3,3))
hist(subdatnumcor$salepercamp, col = "#ece7f2", xlab = "salepercamp", main = "Hist
ogram of salepercamp")
hist(subdatnumcor$salepertrans, col = "#ece7f2", xlab = "salepertrans", main = "Hi
stogram of salepertrans")
hist(subdatnumcor$MED_HOME, col = "#ece7f2", xlab = "MED_HOME", main = "Histogram
of MED_HOME")
boxplot(subdatnumcor$salepercamp, col = "#ece7f2", main = "Boxplot of salepercamp"
)
boxplot(subdatnumcor$salepertrans, col = "#ece7f2", main = "Boxplot of salepertran
s")
boxplot(subdatnumcor$MED_HOME, col = "#ece7f2", main = "Boxplot of MED_HOME")
par(mfrow=c(1,1))

par(mfrow=c(3,3))
hist(subdatnumcor$PRE2009SALES, col = "#ece7f2", xlab = "PRE2009SALES", main = "Hi
stogram of PRE2009SALES")
hist(subdatnumcor$PRE2009TRANSACTIONS, col = "#ece7f2", xlab = "PRE2009TRANSACTION
S", main = "Histogram of PRE2009TRANSACTIONS")
hist(subdatnumcor$cum15QTY, col = "#ece7f2", xlab = "cum15QTY", main = "Histogram
of cum15QTY")
boxplot(subdatnumcor$PRE2009SALES, col = "#ece7f2", main = "Boxplot of PRE2009SALE
S")
boxplot(subdatnumcor$PRE2009TRANSACTIONS, col = "#ece7f2", main = "Boxplot of PRE2
009TRANSACTIONS")
boxplot(subdatnumcor$cum15QTY, col = "#ece7f2", main = "Boxplot of cum15QTY")
par(mfrow=c(1,1))

par(mfrow=c(3,3))
hist(subdatnumcor$QTY15, col = "#ece7f2", xlab = "QTY15", main = "Histogram of QTY
15")
hist(subdatnumcor$cum15TOTAMT, col = "#ece7f2", xlab = "cum15TOTAMT", main = "Hist
ogram of cum15TOTAMT")
```

```
hist(subdatnumcor$TOTAMT15, col = "#ece7f2", xlab = "TOTAMT15", main = "Histogram
of TOTAMT15")
boxplot(subdatnumcor$QTY15, col = "#ece7f2", main = "Boxplot of QTY15")
boxplot(subdatnumcor$cum15TOTAMT, col = "#ece7f2", main = "Boxplot of cum15TOTAMT"
)
boxplot(subdatnumcor$TOTAMT15, col = "#ece7f2", main = "Boxplot of TOTAMT15")
par(mfrow=c(1,1))

par(mfrow=c(2,2))
hist(subdatnumcor$TOTAL_MAIL_15, col = "#ece7f2", xlab = "TOTAL_MAIL_15", main = "
Histogram of TOTAL_MAIL_15")
hist(subdatnumcor$REVPERCUST, col = "#ece7f2", xlab = "REVPERCUST", main = "Histog
ram of REVPERCUST")
boxplot(subdatnumcor$TOTAL_MAIL_15, col = "#ece7f2", main = "Boxplot of TOTAL_MAIL
_15")
boxplot(subdatnumcor$REVPERCUST, col = "#ece7f2", main = "Boxplot of REVPERCUST")
par(mfrow=c(1,1))

par(mfrow=c(2,2))
hist(subdatnumcor$SUM_MAIL_15, col = "#ece7f2", xlab = "SUM_MAIL_15", main = "Hist
ogram of SUM_MAIL_15")
hist(subdatnumcor$INC_SCS_AMT_V4, col = "#ece7f2", xlab = "INC_SCS_AMT_V4", main =
"Histogram of INC_SCS_AMT_V4")
boxplot(subdatnumcor$SUM_MAIL_15, col = "#ece7f2", main = "Boxplot of SUM_MAIL_15"
)
boxplot(subdatnumcor$INC_SCS_AMT_V4, col = "#ece7f2", main = "Boxplot of INC_SCS_A
MT_V4")
par(mfrow=c(1,1))

par(mfrow=c(3,3))
hist(subdatnumcor$INC_WOUTSCS_AMT_4, col = "#ece7f2", xlab = "INC_WOUTSCS_AMT_4",
main = "Histogram of INC_WOUTSCS_AMT_4")
hist(subdatnumcor$MED_INC, col = "#ece7f2", xlab = "MED_INC", main = "Histogram of
MED_INC")
hist(subdatnumcor$MED_FAMINCOM, col = "#ece7f2", xlab = "MED_FAMINCOM", main = "Hi
stogram of MED_FAMINCOM")
boxplot(subdatnumcor$INC_WOUTSCS_AMT_4, col = "#ece7f2", main = "Boxplot of INC_WO
UTSCS_AMT_4")
boxplot(subdatnumcor$MED_INC, col = "#ece7f2", main = "Boxplot of MED_INC")
boxplot(subdatnumcor$MED_FAMINCOM, col = "#ece7f2", main = "Boxplot of MED_FAMINCO
M")
par(mfrow=c(1,1))

par(mfrow=c(2,2))
hist(subdatnumcor$P_CAPITA_INCOM, col = "#ece7f2", xlab = "P_CAPITA_INCOM", main =
"Histogram of P_CAPITA_INCOM")
hist(subdatnumcor$AVG_COMMUTETIM, col = "#ece7f2", xlab = "AVG_COMMUTETIM", main =
"Histogram of AVG_COMMUTETIM")
boxplot(subdatnumcor$P_CAPITA_INCOM, col = "#ece7f2", main = "Boxplot of P_CAPITA_
INCOM")
boxplot(subdatnumcor$AVG_COMMUTETIM, col = "#ece7f2", main = "Boxplot of AVG_COMMU
TETIM")
par(mfrow=c(1,1))

par(mfrow=c(3,3))
hist(subdatnumcor$CUR_EST_MED_INC, col = "#ece7f2", xlab = "CUR_EST_MED_INC", main
= "Histogram of CUR_EST_MED_INC")
hist(subdatnumcor$STATE_INC_INDEX, col = "#ece7f2", xlab = "STATE_INC_INDEX", main
= "Histogram of STATE_INC_INDEX")
```

```
hist(subdatnumcor$STATE_INC_DECILES, col = "#ece7f2", xlab = "STATE_INC_DECILES",
main = "Histogram of STATE_INC_DECILES")
boxplot(subdatnumcor$CUR_EST_MED_INC, col = "#ece7f2", main = "Boxplot of CUR_EST_
MED_INC")
boxplot(subdatnumcor$STATE_INC_INDEX, col = "#ece7f2", main = "Boxplot of STATE_IN
C_INDEX")
boxplot(subdatnumcor$STATE_INC_DECILES, col = "#ece7f2", main = "Boxplot of STATE_
INC_DECILES")
par(mfrow=c(1,1))

par(mfrow=c(3,3))
hist(subdatnumcor$EXAGE, col = "#ece7f2", xlab = "EXAGE", main = "Histogram of EXA
GE")
hist(subdatnumcor$LOR1, col = "#ece7f2", xlab = "LOR1", main = "Histogram of LOR1"
)
hist(subdatnumcor$NUM_CHILD, col = "#ece7f2", xlab = "NUM_CHILD", main = "Histogra
m of NUM_CHILD")
boxplot(subdatnumcor$EXAGE, col = "#ece7f2", main = "Boxplot of EXAGE")
boxplot(subdatnumcor$LOR1, col = "#ece7f2", main = "Boxplot of LOR1")
boxplot(subdatnumcor$NUM_CHILD, col = "#ece7f2", main = "Boxplot of NUM_CHILD")
par(mfrow=c(1,1))

par(mfrow=c(2,2))
hist(subdatnumcor$NUMBADLT, col = "#ece7f2", xlab = "NUMBADLT", main = "Histogram
of NUMBADLT")
hist(subdatnumcor$CNTY_INC, col = "#ece7f2", xlab = "CNTY_INC", main = "Histogram
of CNTY_INC")
boxplot(subdatnumcor$NUMBADLT, col = "#ece7f2", main = "Boxplot of NUMBADLT")
boxplot(subdatnumcor$CNTY_INC, col = "#ece7f2", main = "Boxplot of CNTY_INC")
par(mfrow=c(1,1))

#Outlier Analysis

quantile(train$salepercamp, c(.01, .05, .95, .99))
quantile(train$REVPERCUST, c(.01, .05, .95, .99))
quantile(train$PRE2009SALES, c(.01, .05, .95, .99))
quantile(train$PRE2009TRANSACTIONS, c(.01, .05, .95, .99))
quantile(train$cum15QTY, c(.01, .05, .95, .99))
quantile(train$QTY15, c(.01, .05, .95, .99))
quantile(train$cum15TOTAMT, c(.01, .05, .95, .99))
quantile(train$TOTAMT15, c(.01, .05, .95, .99))
quantile(train$SUM_MAIL_15, c(.01, .05, .95, .99))
quantile(train$TOTAL_MAIL_15, c(.01, .05, .95, .99))
quantile(train$salepertrans, c(.01, .05, .95, .99))
quantile(train$INC_SCS_AMT_V4, c(.01, .05, .95, .99))
quantile(train$INC_WOUTSCS_AMT_4, c(.01, .05, .95, .99))
quantile(train$MED_INC, c(.01, .05, .95, .99))
quantile(train$MED_FAMINCOM, c(.01, .05, .95, .99))
quantile(train$P_CAPITA_INCOM, c(.01, .05, .95, .99))
quantile(train$AVG_COMMUTETIM, c(.01, .05, .95, .99))
quantile(train$MED_HOME, c(.01, .05, .95, .99))
quantile(train$CUR_EST_MED_INC, c(.01, .05, .95, .99))
quantile(train$STATE_INC_INDEX, c(.01, .05, .95, .99))
quantile(train$STATE_INC_DECILES, c(.01, .05, .95, .99))
quantile(train$EXAGE, c(.01, .05, .95, .99))
quantile(train$LOR1, c(.01, .05, .95, .99))
quantile(train$NUM_CHILD, c(.01, .05, .95, .99))
quantile(train$NUMBADLT, c(.01, .05, .95, .99))
quantile(train$CNTY_INC, c(.01, .05, .95, .99))
```

```
summary(train)

#Boxplots for Numeric Variables by RESPONSE16

A1<-ggplot(train, aes(x=RESPONSE16, y= salepercamp)) +
  geom_boxplot(fill="#045a8d",notch=FALSE) +
  labs(title="Distribution of salepercamp") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))
A1

ggplot(train, aes(x=RESPONSE16, y= REVPERCUST)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of REVPERCUST") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

A2<-ggplot(train, aes(x=RESPONSE16, y= PRE2009SALES)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of PRE2009SALES") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))
A2

A3<-ggplot(train, aes(x=RESPONSE16, y= PRE2009TRANSACTIONS)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of PRE2009TRANSACTIONS") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))
A3

A4<-ggplot(train, aes(x=RESPONSE16, y= cum15QTY)) +
  geom_boxplot(fill="#045a8d",notch=FALSE) +
  labs(title="Distribution of cum15QTY") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))
A4

ggplot(train, aes(x=RESPONSE16, y= QTY15)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of QTY15") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= cum15TOTAMT)) +
  geom_boxplot(fill="#045a8d",notch=FALSE) +
  labs(title="Distribution of cum15TOTAMT") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= TOTAMT15)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of TOTAMT15") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= SUM_MAIL_15)) +
  geom_boxplot(fill="#045a8d",notch=FALSE) +
  labs(title="Distribution of SUM_MAIL_15") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

A5<-ggplot(train, aes(x=RESPONSE16, y= TOTAL_MAIL_15)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of TOTAL_MAIL_15") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))
A5
```

```
ggplot(train, aes(x=RESPONSE16, y= salepertrans)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of salepertrans") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= INC_SCS_AMT_V4)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of INC_SCS_AMT_V4") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

A6<-ggplot(train, aes(x=RESPONSE16, y= INC_WOUTSCS_AMT_4)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of INC_WOUTSCS_AMT_4") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))
A6

ggplot(train, aes(x=RESPONSE16, y= MED_INC)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of MED_INC") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= MED_FAMINCOM)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of MED_FAMINCOM") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= P_CAPITA_INCOM)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of P_CAPITA_INCOM") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= AVG_COMMUTETIM)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of AVG_COMMUTETIM") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

A7<-ggplot(train, aes(x=RESPONSE16, y= MED_HOME)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of MED_HOME") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))
A7

ggplot(train, aes(x=RESPONSE16, y= CUR_EST_MED_INC)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of CUR_EST_MED_INC") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= STATE_INC_INDEX)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of STATE_INC_INDEX") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= STATE_INC_DECILES)) +
  geom_boxplot(fill="#045a8d",notch=FALSE) +
  labs(title="Distribution of STATE_INC_DECILES") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= EXAGE)) +
```

```
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of EXAGE") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= LOR1)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of LOR1") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= NUM_CHILD)) +
  geom_boxplot(fill="#045a8d",notch=FALSE) +
  labs(title="Distribution of NUM_CHILD") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= NUMBADLT)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of NUMBADLT") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train, aes(x=RESPONSE16, y= CNTY_INC)) +
  geom_boxplot(fill="#045a8d",notch=TRUE) +
  labs(title="Distribution of CNTY_INC") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

multiplot(A5,A7, cols=2)

#Correlation Matrix
library(GGally) #Data Visualization
require(lattice)
library(ggcorrplot) #Data Visualization
library(corrplot) #Data Visualization
par(mfrow=c(1,1))
corr <- round(cor(subdatnumcor),2)
ggcorrplot(corr, outline.col = "white", ggtheme = ggplot2::theme_gray,
           colors = c("#E46726", "white", "#6D9EC1"),lab = TRUE)
par(mfrow=c(1,1))

###################### EDA Categorical ################################
library(ggplot2)
ggplot(train) +
  geom_bar( aes(RESPONSE16) ) +
  ggtitle("RESPONSE16") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

library(ggplot2)
ggplot(train) +
  geom_bar( aes(ZIP) ) +
  ggtitle("ZIP") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(CHANNEL_ACQUISITION) ) +
  ggtitle("CHANNEL_ACQUISITION") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(DEBIT_CC) ) +
  ggtitle("DEBIT_CC") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))
```

```
ggplot(train) +
  geom_bar( aes(MAJOR_CC) ) +
  ggtitle("MAJOR_CC") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(COMPUTER_ELECTRONIC) ) +
  ggtitle("COMPUTER_ELECTRONIC") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(INC_WIOUTSCS_V4) ) +
  ggtitle("INC_WIOUTSCS_V4") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(INC_WITHSCS_V4) ) +
  ggtitle("INC_WITHSCS_V4") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(FIPSCNTY) ) +
  ggtitle("FIPSCNTY") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ADULT1_G) ) +
  ggtitle("ADULT1_G") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(MARRIED) ) +
  ggtitle("MARRIED") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ETHNIC_MATCH) ) +
  ggtitle("ETHNIC_MATCH") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(HOMEOWNR) ) +
  ggtitle("HOMEOWNR") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ADD_TYPE) ) +
  ggtitle("ADD_TYPE") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(DUS) ) +
  ggtitle("DUS") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(IND_DMR) ) +
  ggtitle("IND_DMR") +
```

```
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(OCCUPATION_GROUP) ) +
  ggtitle("OCCUPATION_GROUP") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(IND_ED) ) +
  ggtitle("IND_ED") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(PRESCHLD) ) +
  ggtitle("PRESCHLD") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(MAILPREF) ) +
  ggtitle("MAILPREF") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(CHANNEL_DOMINANCE) ) +
  ggtitle("CHANNEL_DOMINANCE") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(SALES) ) +
  ggtitle("SALES") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ZONLINE) ) +
  ggtitle("ZONLINE") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ZMOB) ) +
  ggtitle("ZMOB") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ZPRCHPHN) ) +
  ggtitle("ZPRCHPHN") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ZMOBMULT) ) +
  ggtitle("ZMOBMULT") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ZHMDECOR) ) +
  ggtitle("ZHMDECOR") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ZHOMEENT) ) +
```

```
  ggtitle("ZHOMEENT") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

ggplot(train) +
  geom_bar( aes(ZKITCHEN) ) +
  ggtitle("ZKITCHEN") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5))

###Crosstabs
library(gmodels)
library(gridExtra)
attach(train)

#ZIP
CrossTable(RESPONSE16,ZIP, prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chisq=FALS
E)
l <- ggplot(train, aes(ZIP ,fill = RESPONSE16)) + geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ZIP,mean)

#ZIP
CrossTable(RESPONSE16,ZIP, prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chisq=FALS
E)
l <- ggplot(train, aes(ZIP ,fill = RESPONSE16)) + geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ZIP,mean)

A1 <- ggplot(train, aes(x = ZIP, fill = RESPONSE16)) + geom_bar(position = 'fill')
+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal() + coord_flip()
grid.arrange(l,A1, nrow = 2, top = textGrob("ZIP",gp=gpar(fontsize=15,font=2)))

#INC_WIOUTSCS_V4
CrossTable(RESPONSE16, INC_WIOUTSCS_V4 , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, p
rop.chisq=FALSE)
l <- ggplot(train, aes(INC_WIOUTSCS_V4 ,fill = RESPONSE16))+ geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ INC_WIOUTSCS_V4 ,mean)

A10 <- ggplot(train, aes(x = INC_WIOUTSCS_V4 , fill = RESPONSE16)) + geom_bar(posi
tion = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+theme_minimal() + coord_flip()
grid.arrange(l,A10, nrow = 2,top = textGrob("INC_WIOUTSCS_V4",gp=gpar(fontsize=15,
font=2)))

multiplot(A2, A10, cols=1)
#DEBIT_CC
CrossTable(RESPONSE16,DEBIT_CC, prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chisq
=FALSE)
l <- ggplot(train, aes(DEBIT_CC ,fill = RESPONSE16)) + geom_bar(position = 'fill')
+
```

```
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$DEBIT_CC,mean)

A1 <- ggplot(train, aes(x = DEBIT_CC, fill = RESPONSE16)) + geom_bar(position = 'f
ill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal() + coord_flip()
grid.arrange(l,A1, nrow = 2, top = textGrob("DEBIT_CC",gp=gpar(fontsize=15,font=2)
))

#MAJOR_CC
CrossTable(RESPONSE16,MAJOR_CC, prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chisq
=FALSE)
l <- ggplot(train, aes(MAJOR_CC ,fill = RESPONSE16)) + geom_bar(position = 'fill')
+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$MAJOR_CC,mean)

A1 <- ggplot(train, aes(x = MAJOR_CC, fill = RESPONSE16)) + geom_bar(position = 'f
ill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal() + coord_flip()
grid.arrange(l,A1, nrow = 2, top = textGrob("MAJOR_CC",gp=gpar(fontsize=15,font=2)
))

#COMPUTER_ELECTRONIC
CrossTable(RESPONSE16,COMPUTER_ELECTRONIC, prop.r=FALSE, prop.c=TRUE, prop.t=TRUE,
prop.chisq=FALSE)
l <- ggplot(train, aes(COMPUTER_ELECTRONIC ,fill = RESPONSE16)) + geom_bar(positio
n = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$COMPUTER_ELECTRONIC,mean)

A1 <- ggplot(train, aes(x = COMPUTER_ELECTRONIC, fill = RESPONSE16)) + geom_bar(po
sition = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal() + coord_flip()
grid.arrange(l,A1, nrow = 2, top = textGrob("COMPUTER_ELECTRONIC",gp=gpar(fontsize
=15,font=2)))

#INC_WITHSCS_V4
CrossTable(RESPONSE16, INC_WITHSCS_V4 , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, pr
op.chisq=FALSE)
l <- ggplot(train, aes(INC_WITHSCS_V4 ,fill = RESPONSE16))+ geom_bar(position = 'f
ill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ INC_WITHSCS_V4 ,mean)

A3 <- ggplot(train, aes(x = INC_WITHSCS_V4 , fill = RESPONSE16)) + geom_bar(positi
on = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+theme_minimal() + coord_flip()
```

```
grid.arrange(l,A3, nrow = 2,top = textGrob("INC_WITHSCS_V4",gp=gpar(fontsize=15,fo
nt=2)))

#FIPSCNTY
CrossTable(RESPONSE16, FIPSCNTY , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(FIPSCNTY ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ FIPSCNTY ,mean)

A4 <- ggplot(train, aes(x = FIPSCNTY , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+theme_minimal() + coord_flip()
grid.arrange(l,A4, nrow = 2,top = textGrob("FIPSCNTY ",gp=gpar(fontsize=15,font=2)
))

#ADULT1_G
CrossTable(RESPONSE16, ADULT1_G , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(ADULT1_G ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ADULT1_G ,mean)

A4 <- ggplot(train, aes(x = ADULT1_G , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+theme_minimal() + coord_flip()
grid.arrange(l,A4, nrow = 2,top = textGrob("ADULT1_G ",gp=gpar(fontsize=15,font=2)
))

#MARRIED
CrossTable(RESPONSE16,MARRIED, prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chisq=
FALSE)
l <- ggplot(train, aes(MARRIED ,fill = RESPONSE16)) + geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$MARRIED,mean)

B1 <- ggplot(train, aes(x = MARRIED, fill = RESPONSE16)) + geom_bar(position = 'fi
ll')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal() + coord_flip()
grid.arrange(l,B1, nrow = 2, top = textGrob("MARRIED",gp=gpar(fontsize=15,font=2))
)

#ETHNIC_MATCH
CrossTable(RESPONSE16,ETHNIC_MATCH, prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.c
hisq=FALSE)
l <- ggplot(train, aes(ETHNIC_MATCH ,fill = RESPONSE16)) + geom_bar(position = 'fi
ll')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ETHNIC_MATCH,mean)
```

```
B1 <- ggplot(train, aes(x = ETHNIC_MATCH, fill = RESPONSE16)) + geom_bar(position
= 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal() + coord_flip()
grid.arrange(l,B1, nrow = 2, top = textGrob("ETHNIC_MATCH",gp=gpar(fontsize=15,fon
t=2)))

#HOMEOWNR
CrossTable(RESPONSE16, HOMEOWNR , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(HOMEOWNR ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ HOMEOWNR ,mean)

B2 <- ggplot(train, aes(x = HOMEOWNR , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+theme_minimal() + coord_flip()
grid.arrange(l,B2, nrow = 2,top = textGrob("HOMEOWNR",gp=gpar(fontsize=15,font=2))
)

#ADD_TYPE
CrossTable(RESPONSE16, ADD_TYPE , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(ADD_TYPE ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ADD_TYPE ,mean)

B2 <- ggplot(train, aes(x = ADD_TYPE , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+theme_minimal() + coord_flip()
grid.arrange(l,B2, nrow = 2,top = textGrob("ADD_TYPE",gp=gpar(fontsize=15,font=2))
)

#IND_DMR
CrossTable(RESPONSE16, IND_DMR , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chis
q=FALSE)
l <- ggplot(train, aes(IND_DMR ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ IND_DMR ,mean)

B2 <- ggplot(train, aes(x = IND_DMR , fill = RESPONSE16)) + geom_bar(position = 'f
ill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+theme_minimal() + coord_flip()
grid.arrange(l,B2, nrow = 2,top = textGrob("IND_DMR",gp=gpar(fontsize=15,font=2)))

#DUS
CrossTable(RESPONSE16, DUS , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chisq=FA
LSE)
l <- ggplot(train, aes(DUS ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ DUS ,mean)
```

```
B2 <- ggplot(train, aes(x = DUS , fill = RESPONSE16)) + geom_bar(position = 'fill'
)+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+theme_minimal() + coord_flip()
grid.arrange(l,B2, nrow = 2,top = textGrob("DUS",gp=gpar(fontsize=15,font=2)))

#OCCUPATION_GROUP
CrossTable(RESPONSE16, OCCUPATION_GROUP , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE,
prop.chisq=FALSE)
l <- ggplot(train, aes(OCCUPATION_GROUP ,fill = RESPONSE16))+ geom_bar(position =
'fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ OCCUPATION_GROUP ,mean)

B2 <- ggplot(train, aes(x = OCCUPATION_GROUP , fill = RESPONSE16)) + geom_bar(posi
tion = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+theme_minimal() + coord_flip()
grid.arrange(l,B2, nrow = 2,top = textGrob("OCCUPATION_GROUP",gp=gpar(fontsize=15,
font=2)))

#IND_ED
CrossTable(RESPONSE16, IND_ED , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chisq
=FALSE)
l <- ggplot(train, aes(DUS ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ IND_ED ,mean)

B2 <- ggplot(train, aes(x = IND_ED , fill = RESPONSE16)) + geom_bar(position = 'fi
ll')+
  scale_fill_manual(values=c("#9ecae1", "#3182bd"))+theme_minimal() + coord_flip()
grid.arrange(l,B2, nrow = 2,top = textGrob("IND_ED",gp=gpar(fontsize=15,font=2)))

#MAILPREF
CrossTable(RESPONSE16, MAILPREF , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(MAILPREF ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ MAILPREF ,mean)

B3 <- ggplot(train, aes(x = MAILPREF , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+theme_minimal() + coord_flip()
grid.arrange(l,B3, nrow = 2,top = textGrob("MAILPREF ",gp=gpar(fontsize=15,font=2)
))

#PRESCHLD
CrossTable(RESPONSE16, PRESCHLD , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(PRESCHLD ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ PRESCHLD ,mean)
```

```
B3 <- ggplot(train, aes(x = PRESCHLD , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+theme_minimal() + coord_flip()
grid.arrange(l,B3, nrow = 2,top = textGrob("PRESCHLD ",gp=gpar(fontsize=15,font=2)
))

#CHANNEL_DOMINANCE
CrossTable(RESPONSE16, CHANNEL_DOMINANCE , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE,
prop.chisq=FALSE)
l <- ggplot(train, aes(CHANNEL_DOMINANCE ,fill = RESPONSE16))+ geom_bar(position =
'fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ CHANNEL_DOMINANCE ,mean)

B3 <- ggplot(train, aes(x = CHANNEL_DOMINANCE , fill = RESPONSE16)) + geom_bar(pos
ition = 'fill')+
  scale_fill_manual(values=c("#9ecae1", "#08306b"))+theme_minimal() + coord_flip()
grid.arrange(l,B3, nrow = 2,top = textGrob("CHANNEL_DOMINANCE ",gp=gpar(fontsize=1
5,font=2)))

#SALES
CrossTable(RESPONSE16, SALES , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chisq=
FALSE)
l <- ggplot(train, aes(SALES ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ SALES ,mean)

A2 <- ggplot(train, aes(x = SALES , fill = RESPONSE16)) + geom_bar(position = 'fil
l')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+theme_minimal() + coord_flip()
grid.arrange(l,A2, nrow = 2,top = textGrob("SALES",gp=gpar(fontsize=15,font=2)))

#ZONLINE
CrossTable(RESPONSE16, ZONLINE , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chis
q=FALSE)
l <- ggplot(train, aes(ZONLINE ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ZONLINE ,mean)

A2 <- ggplot(train, aes(x = ZONLINE , fill = RESPONSE16)) + geom_bar(position = 'f
ill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+theme_minimal() + coord_flip()
grid.arrange(l,A2, nrow = 2,top = textGrob("ZONLINE",gp=gpar(fontsize=15,font=2)))

#ZMOB
CrossTable(RESPONSE16, ZMOB , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chisq=F
ALSE)
l <- ggplot(train, aes(ZMOB ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ZMOB ,mean)
```

```
A2 <- ggplot(train, aes(x = ZMOB , fill = RESPONSE16)) + geom_bar(position = 'fill
')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+theme_minimal() + coord_flip()
grid.arrange(l,A2, nrow = 2,top = textGrob("ZMOB",gp=gpar(fontsize=15,font=2)))

#ZPRCHPHN
CrossTable(RESPONSE16, ZPRCHPHN , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(ZPRCHPHN ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ZPRCHPHN ,mean)

A2 <- ggplot(train, aes(x = ZPRCHPHN , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+theme_minimal() + coord_flip()
grid.arrange(l,A2, nrow = 2,top = textGrob("ZPRCHPHN",gp=gpar(fontsize=15,font=2))
)

#ZMOBMULT
CrossTable(RESPONSE16, ZMOBMULT , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(ZMOBMULT ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ZMOBMULT ,mean)

A2 <- ggplot(train, aes(x = ZMOBMULT , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+theme_minimal() + coord_flip()
grid.arrange(l,A2, nrow = 2,top = textGrob("ZMOBMULT",gp=gpar(fontsize=15,font=2))
)

#ZHMDECOR
CrossTable(RESPONSE16, ZHMDECOR , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(ZHMDECOR ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ZHMDECOR ,mean)

A2 <- ggplot(train, aes(x = ZHMDECOR , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+theme_minimal() + coord_flip()
grid.arrange(l,A2, nrow = 2,top = textGrob("ZHMDECOR",gp=gpar(fontsize=15,font=2))
)

#ZHOMEENT
CrossTable(RESPONSE16, ZHOMEENT , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(ZHOMEENT ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ZHOMEENT ,mean)
```

```
A2 <- ggplot(train, aes(x = ZHOMEENT , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+theme_minimal() + coord_flip()
grid.arrange(l,A2, nrow = 2,top = textGrob("ZHOMEENT",gp=gpar(fontsize=15,font=2))
)

#ZKITCHEN
CrossTable(RESPONSE16, ZKITCHEN , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(ZKITCHEN ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ZKITCHEN ,mean)

A2 <- ggplot(train, aes(x = ZKITCHEN , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+theme_minimal() + coord_flip()
grid.arrange(l,A2, nrow = 2,top = textGrob("ZKITCHEN",gp=gpar(fontsize=15,font=2))
)

#ZPRCHONL
CrossTable(RESPONSE16, ZPRCHONL , prop.r=FALSE, prop.c=TRUE, prop.t=TRUE, prop.chi
sq=FALSE)
l <- ggplot(train, aes(ZPRCHONL ,fill = RESPONSE16))+ geom_bar(position = 'fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+
  theme_minimal()
l <- l + geom_histogram(stat="count")
tapply(as.numeric(train$RESPONSE16) - 1 , train$ ZPRCHONL ,mean)

A2 <- ggplot(train, aes(x = ZPRCHONL , fill = RESPONSE16)) + geom_bar(position = '
fill')+
  scale_fill_manual(values=c("#a6bddb", "#2b8cbe"))+theme_minimal() + coord_flip()
grid.arrange(l,A2, nrow = 2,top = textGrob("ZPRCHONL",gp=gpar(fontsize=15,font=2))
)

#Prevalance Statistic for Responses16
#Train
table(train$RESPONSE16) #1008 bought, 9438 did not buy
1008/(9438 + 1008) #0.09649627 bought
9438/(9438 + 1008) #0.9035037 did not buy

#validation
table(validation$RESPONSE16) #432 bought, 4044 did not buy
432/(4044 + 432) #0.09651475 bought
4044/(4044 + 432) #0.9034853 did not buy

str(train)

######################################### Build Models ##############################
#########

#############################################################################
######## Step 4 build models - Logistic Regression Model #############
#############################################################################

library(glm2)
library(car)
```

```
detach(package:ModelMetrics)

#Full Model for Variable Selection & Baseline
model.logfull <- glm(RESPONSE16 ~ ., data = train, family=binomial("logit"))
varImp(model.logfull)

#Stepwise Regression for Variable Selection
model.lower = glm(RESPONSE16 ~ 1, train, family = binomial(link="logit"))
model.logfull <- glm(RESPONSE16 ~ ., train, family=binomial("logit"))
stepAIC(model.lower, scope = list(upper=model.logfull), direction="both", validati
on="Chisq", data=balanced.data)

#Original
#PRE2009TRANSACTIONS + TOTAL_MAIL_15 +
 # QTY15 + CHANNEL_ACQUISITION + salepercamp + ADULT1_G + PRE2009SALES +
  #MED_HOME + ZPRCHONL + cum15QTY + CHANNEL_DOMINANCE + INC_WOUTSCS_AMT_4 +
  #HOMEOWNR + AVG_COMMUTETIM + ZHMDECOR

model.log2 <- glm(RESPONSE16 ~ PRE2009TRANSACTIONS + TOTAL_MAIL_15 +
                    QTY15 + CHANNEL_ACQUISITION + salepercamp + ADULT1_G + PRE2009
SALES +
                    MED_HOME + ZPRCHONL + cum15QTY + CHANNEL_DOMINANCE + INC_WOUTS
CS_AMT_4 +
                    HOMEOWNR + AVG_COMMUTETIM + ZHMDECOR,data = train, family=bino
mial("logit"))

summary(model.log2)
hoslem.test(train$RESPONSE16, fitted(model.log2)) #check p-value and is small, rej
ect null hypothesis and statistically significant model
Anova(model.log2, type="II", validation="Wald")
varImp(model.log2)
#nagelkerke(model.log2)
vif(model.log2)

#Performance Metrics
AIC(model.log2) #6100.637
BIC(model.log2) #6267.478

##Performance on validation Set
glm.pred <- predict(model.log2, validation, type="response")
hist(glm.pred)#Histogram of predicted probabilities

#Confusion Matrix
glm.pred  <- ifelse(glm.pred  > 0.5,1,0)
xtab.log1=table(glm.pred, validation$RESPONSE16)
confusionMatrix(xtab.log1, positive = "1") #0.9015

#Plot Curves
require(PRROC)
prob <- predict(model.log2, newdata=validation, type="response")

fg <- prob[validation$RESPONSE16 == 1]
bg <- prob[validation$RESPONSE16 == 0]

#ROC Curve
roc <- roc.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
plot(roc)

#Precision Recall Curve
```

```
pr <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
plot(pr)

#Compute area under ROC curve
roc <- roc.curve( fg, bg );
print( roc ) #0.7151599

#Compute area under PR curve
pr <- pr.curve( fg, bg )
print( pr ) #0.2142101

#ROC Curve (Alternative)
#library(ROCR)
#prob <- predict(model.log2, newdata=validation, type="response")
#pred <- prediction(prob, validation$RESPONSE16)
#perf <- performance(pred, measure = "tpr", x.measure = "fpr")
#plot(perf,lwd=2, col="#3182bd", main="ROC Curve", colorize=TRUE)
#abline(a=0, b=1)

#AUC
#perf_auc <- performance(pred, measure = "auc")
#perf_auc <- perf_auc@y.values[[1]]
#perf_auc #0.7151599

## Evaluate on All data in subset
glm.pred_all<- predict(model.log2, subdat2, type = "response")
hist(glm.pred_all)

pred2df <- as.data.frame(glm.pred_all)
data_all <- cbind(subdat2,glm.pred_all)
str(data_all)

#Average Sales Per Transactions
mean(data_all$salepertrans) #Average subset is $254.8435
mean(data_all$salepertrans [data_all$glm.pred_all>0.70]) #70% probability is 318.3
577

#Average Revenue per Customer for All
mean(data_all$REVPERCUST) #Average subset is $271.1046

#Targeted Customers with New Model
sum(data_all$glm.pred_all>0.65) #53
sum(data_all$glm.pred_all>0.5) #100
sum(data_all$glm.pred_all>0.4) #189
sum(data_all$glm.pred_all>0.3) #390
sum(data_all$glm.pred_all>0.2) #1083
sum(data_all$glm.pred_all>0.15) #2113
sum(data_all$glm.pred_all>0.10) #4558
sum(data_all$glm.pred_all>0.05) #11441

#Average Revenue per Customer for New Model
mean(data_all$REVPERCUST [data_all$glm.pred_all>0.65]) #3093.568
mean(data_all$REVPERCUST [data_all$glm.pred_all>0.5]) #2690.291
mean(data_all$REVPERCUST [data_all$glm.pred_all>0.4]) #2449.486
mean(data_all$REVPERCUST [data_all$glm.pred_all>0.3]) #1953.931
mean(data_all$REVPERCUST [data_all$glm.pred_all>0.2]) #1265.134
mean(data_all$REVPERCUST [data_all$glm.pred_all>0.15]) #963.7352
mean(data_all$REVPERCUST [data_all$glm.pred_all>0.10]) #634.8024
mean(data_all$REVPERCUST [data_all$glm.pred_all>0.05]) #333.3991
```

```
str(glm.pred_all)

########################################################################
######## Step 5 LDA model ##############################
########################################################################
detach(package:ModelMetrics)

###LDA###
model.lda2 <- lda(RESPONSE16 ~ PRE2009TRANSACTIONS + TOTAL_MAIL_15 +
                    QTY15 + CHANNEL_ACQUISITION + salepercamp + ADULT1_G + PRE2009
SALES +
                    MED_HOME + ZPRCHONL + cum15QTY + CHANNEL_DOMINANCE + INC_WOUTS
CS_AMT_4 +
                    HOMEOWNR + AVG_COMMUTETIM + ZHMDECOR,train, family=binomial("l
ogit"))

##Performance on validation Set
lda.pred <- predict(model.lda2, validation)$posterior[,2]
hist(lda.pred)

#Confusion Matrix
lda.pred_cm  <- ifelse(lda.pred  > 0.5,1,0)
xtab.lda1=table(lda.pred_cm, validation$RESPONSE16)
confusionMatrix(xtab.lda1, positive = "1") #0.8975

#Plot Curves
require(PRROC)
prob <- predict(model.lda2, validation)$posterior[,2]

fg <- prob[validation$RESPONSE16 == 1]
bg <- prob[validation$RESPONSE16 == 0]

#ROC Curve
roc <- roc.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
plot(roc)

#Precision Recall Curve
pr <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
plot(pr)

#Compute area under ROC curve
roc <- roc.curve( fg, bg );
print( roc ) #0.7193693

#Compute area under PR curve
pr <- pr.curve( fg, bg )
print( pr ) #0.2198539

#ROC Curve (Alternative)
#library(ROCR)

#validation.lda <-  predict(model.lda2, validation)$posterior
#pred <- prediction(validation.lda[,2], validation$RESPONSE16)
#perf <- performance(pred, "tpr", "fpr")
#plot(perf,lwd=2, col="#3182bd", main="ROC Curve", colorize=TRUE)
#abline(a=0, b=1)

#AUC
```

```
#perf_auc <- performance(pred, measure = "auc")
#perf_auc <- perf_auc@y.values[[1]]
#perf_auc

## Evaluate on All data in subset
lda.pred_all <- predict(model.lda2, subdat2)$posterior[,2]
hist(lda.pred_all)

pred2df <- as.data.frame(lda.pred_all)
data_all2 <- cbind(subdat2,lda.pred_all)
str(data_all2)

#Average Revenue per Customer for All
mean(data_all2$REVPERCUST) #Average subset is $271.1046

#Targeted Customers with New Model
sum(data_all2$lda.pred_all>0.65) #142
sum(data_all2$lda.pred_all>0.5) #268
sum(data_all2$lda.pred_all>0.4) #408
sum(data_all2$lda.pred_all>0.3) #678
sum(data_all2$lda.pred_all>0.2) #1283
sum(data_all2$lda.pred_all>0.15) #2065
sum(data_all2$lda.pred_all>0.10) #3708
sum(data_all2$lda.pred_all>0.05) #8771

#Average Revenue per Customer for New Model
mean(data_all2$REVPERCUST[data_all2$lda.pred_all>0.65]) #2845.481
mean(data_all2$REVPERCUST[data_all2$lda.pred_all>0.5]) #2416.554
mean(data_all2$REVPERCUST[data_all2$lda.pred_all>0.4]) #1990.168
mean(data_all2$REVPERCUST[data_all2$lda.pred_all>0.3]) #1673.998
mean(data_all2$REVPERCUST[data_all2$lda.pred_all>0.2]) #1287.399
mean(data_all2$REVPERCUST[data_all2$lda.pred_all>0.15]) #1032.211
mean(data_all2$REVPERCUST[data_all2$lda.pred_all>0.10]) #741.666
mean(data_all2$REVPERCUST[data_all2$lda.pred_all>0.05]) #411.6367

#####################################################################
######## Step 6 Random Forest model ################################
#####################################################################

#Balance Imbalanced Data using SMOTE (Synthetic Minority Over-sampling Technique)
library(DMwR)
set.seed(1)
balanced.data <- SMOTE(RESPONSE16 ~., train, perc.over = 100, k = 5, perc.under =
200)
as.data.frame(table(balanced.data$RESPONSE16))
prop.table(table(balanced.data$RESPONSE16))

###Random Forest Model###
library(caret)
detach(package:ModelMetrics)

#Use RF to obtain important variables from subset
set.seed(12)
rf1 <- randomForest(RESPONSE16 ~ .,data=balanced.data,importance=TRUE,ntree=100)
summary(rf1)

##getTree(rf1,1,labelVar=TRUE)
##?getTree
print(rf1)
```

```
plot(rf1) #100
importance(rf1)
importance(rf1)
varImpPlot(rf1)

#Create Random Forest Model with Top 20 Important Variables
set.seed(1)
detach(package:ModelMetrics)
#rf.model <- randomForest(RESPONSE16 ~ PRE2009TRANSACTIONS + ZIP + PRE2009SALES +
SALES + INC_WIOUTSCS_V4 + DUS
               #          + INC_WIOUTSCS_V4 + TOTAL_MAIL_15 + cum15QTY + salepertr
ans + INC_WOUTSCS_AMT_4 + CNTY_INC
                #         + cum15TOTAMT + STATE_INC_INDEX + EXAGE + CUR_EST_MED_IN
C + MED_INC + MED_HOME + REVPERCUST + LOR1
                 #        = balanced.data,importance=TRUE,ntree=100)


#Choose Predictors that Overlapped (stepAIC vs. RF Top 20 Important Variables)
set.seed(1)
detach(package:ModelMetrics)

rf.model <- randomForest(RESPONSE16 ~ PRE2009TRANSACTIONS + TOTAL_MAIL_15 + PRE200
9SALES + MED_HOME
                         + cum15QTY + INC_WIOUTSCS_V4, data = balanced.data,import
ance=TRUE,ntree=100)

summary(rf.model)

##getTree(rf1,1,labelVar=TRUE)
##?getTree
print(rf.model)
plot(rf.model) #100
importance(rf.model)
varImpPlot(rf.model)

##Performance on validation Set
set.seed(1)
prob <- predict(rf.model, newdata = validation, type = "prob")[,2]
hist(prob)#Histogram of predicted probabilities

#Confusion Matrix for Probabilities
library(caret)

glm.pred  <- ifelse(prob  > 0.50,1,0)
xtab.RF1=table(glm.pred, validation$RESPONSE16)
confusionMatrix(xtab.RF1, positive = "1") #0.7339

#Confusion Matrix Detail
confusion.matrix <- table(prob, validation$RESPONSE16)
confusion.matrix

#Plot Curves
require(PRROC)
set.seed(1)

fg <- prob[validation$RESPONSE16 == 1]
bg <- prob[validation$RESPONSE16 == 0]

#ROC Curve
```

```
roc <- roc.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
plot(roc)

#Precision Recall Curve
pr <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
plot(pr)

#Compute area under ROC curve
roc <- roc.curve( fg, bg );
print( roc ) #0.6850298

#Compute area under PR curve
pr <- pr.curve( fg, bg )
print( pr ) #0.1906895

#AUC
#library(ModelMetrics)
#set.seed(1)
#prob <- predict(rf.model, newdata=validation, type='prob')
#auc<-auc(validation$RESPONSE16,prob[,2])
#auc #0.6850298

#ROC Curve
#rf1pdf <- as.data.frame(prob)
#rf1.pred = prediction(prob, validation$RESPONSE16)
#rf1.perf = performance(rf1.pred,"tpr","fpr")
#plot(rf1.perf,main="ROC Curve for Random Forest",col=2,lwd=2)
#abline(a=0,b=1,lwd=2,lty=2,col="gray")

## Evaluate on All data in subset
set.seed(1)
prediction_rf_all<- predict(rf.model, newdata = subdat2, type = "prob")[,2]
hist(prediction_rf_all)

pred2df <- as.data.frame(prediction_rf_all)
data_alldf <- cbind(subdat2,prediction_rf_all)
str(data_alldf)

#Average Sales Per Transactions
mean(data_alldf$salepertrans) #Average subset is $254.8435
mean(data_alldf$salepertrans[data_alldf$prediction_rf_all>0.70]) #70% probability
is 296.3083

#Average Revenue per Customer for All
mean(data_alldf$REVPERCUST) #Average subset is $271.1046

#Targeted Customers with New Model
sum(data_alldf$prediction_rf_all>0.65) #2332
sum(data_alldf$prediction_rf_all>0.5) #4003
sum(data_alldf$prediction_rf_all>0.4) #5353
sum(data_alldf$prediction_rf_all>0.3) #7090
sum(data_alldf$prediction_rf_all>0.2) #9416
sum(data_alldf$prediction_rf_all>0.15) #10706
sum(data_alldf$prediction_rf_all>0.10) #12120
sum(data_alldf$prediction_rf_all>0.05) #13660

#Average Revenue per Customer for New Model
mean(data_alldf$REVPERCUST[data_alldf$prediction_rf_all>0.65]) #699.7214
mean(data_alldf$REVPERCUST[data_alldf$prediction_rf_all>0.5]) #615.9711
```

```
mean(data_alldf$REVPERCUST[data_alldf$prediction_rf_all>0.4]) #537.7053
mean(data_alldf$REVPERCUST[data_alldf$prediction_rf_all>0.3]) #454.294
mean(data_alldf$REVPERCUST[data_alldf$prediction_rf_all>0.2]) #380.1277
mean(data_alldf$REVPERCUST[data_alldf$prediction_rf_all>0.15]) #350.9264
mean(data_alldf$REVPERCUST[data_alldf$prediction_rf_all>0.10]) #321.3439
mean(data_alldf$REVPERCUST[data_alldf$prediction_rf_all>0.05]) #292.8959

str(prediction_rf_all)

########################################## XYZ Didn't Target ###################
#####################################
subdat3 <- subset(subdat, ANY_MAIL_16 == 0)
str(subdat3)

##### Change Variable Types #####
#Factors
subdat2$ZIP <- as.factor(subdat2$ZIP)
subdat3$RESPONSE16 <- as.factor(subdat3$RESPONSE16)
subdat3$CHANNEL_ACQUISITION <- as.factor(subdat3$CHANNEL_ACQUISITION)
subdat3$DEBIT_CC <- as.factor(subdat3$DEBIT_CC)
subdat3$MAJOR_CC <- as.factor(subdat3$MAJOR_CC)
subdat3$COMPUTER_ELECTRONIC <- as.factor(subdat3$COMPUTER_ELECTRONIC)
subdat3$INC_WIOUTSCS_V4 <- as.factor(subdat3$INC_WIOUTSCS_V4)
subdat3$INC_WITHSCS_V4 <- as.factor(subdat3$INC_WITHSCS_V4)
subdat3$FIPSCNTY <- as.factor(subdat3$FIPSCNTY)
subdat3$ADULT1_G <- as.factor(subdat3$ADULT1_G)
subdat3$MARRIED <- as.factor(subdat3$MARRIED)
subdat3$ETHNIC_MATCH <- as.factor(subdat3$ETHNIC_MATCH)
subdat3$HOMEOWNR <- as.factor(subdat3$HOMEOWNR)
subdat3$ADD_TYPE <- as.factor(subdat3$ADD_TYPE)
subdat3$DUS <- as.factor(subdat3$DUS)
subdat3$IND_DMR <- as.factor(subdat3$IND_DMR)
subdat3$OCCUPATION_GROUP <- as.factor(subdat3$OCCUPATION_GROUP)
subdat3$IND_ED <- as.factor(subdat3$IND_ED)
subdat3$PRESCHLD <- as.factor(subdat3$PRESCHLD)
subdat3$MAILPREF <- as.factor(subdat3$MAILPREF)
subdat3$CHANNEL_DOMINANCE <- as.factor(subdat3$CHANNEL_DOMINANCE)
subdat3$SALES <- as.factor(subdat3$SALES)
subdat3$ZONLINE <- as.factor(subdat3$ZONLINE)
subdat3$ZMOB <- as.factor(subdat3$ZMOB)
subdat3$ZPRCHPHN <- as.factor(subdat3$ZPRCHPHN)
subdat3$ZMOBMULT <- as.factor(subdat3$ZMOBMULT)
subdat3$ZHMDECOR <- as.factor(subdat3$ZHMDECOR)
subdat3$ZHOMEENT <- as.factor(subdat3$ZHOMEENT)
subdat3$ZKITCHEN <- as.factor(subdat3$ZKITCHEN)
subdat3$ZPRCHONL <- as.factor(subdat3$ZPRCHONL)

#Numeric
subdat3$EXAGE <- as.numeric(subdat3$EXAGE)
subdat3$LOR1 <- as.numeric(subdat3$LOR1)
subdat3$NUM_CHILD <- as.numeric(subdat3$NUM_CHILD)
subdat3$SUM_MAIL_15 <- as.numeric(subdat3$SUM_MAIL_15)
subdat3$TOTAL_MAIL_15 <- as.numeric(subdat3$TOTAL_MAIL_15)
subdat3$REVPERCUST <- as.numeric(subdat3$REVPERCUST)
subdat3$salepercamp <- as.numeric(subdat3$salepercamp)

str(subdat3)
#Remove Variables
subdat3$ANY_MAIL_16 <- NULL
```

```
################################################################################
####### Step 3 - EDA - Cleaning up of the data for XYZ Didn't Target  ############
################################################################################

str(subdat3)

#Check for Missingness
sum(is.na(subdat3))
sapply(subdat3, function(x) sum(is.na(x)))
aggr_plot <- aggr(subdat3, col=c('#9ecae1','#de2d26'), numbers=TRUE,prop=FALSE, so
rtVars=TRUE, labels=names(subdat3), cex.axis=.5, gap=2, ylab=c("Histogram of missi
ng data","Pattern"))

#Check missing data percentage
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(subdat3,2,pMiss)

# Convert Blanks to NA
subdat3[subdat3 == ""] <- NA

#Impute Factor Variables with Majority Class (except ETHNIC_MATCH)
subdat3$DEBIT_CC[is.na(subdat3$DEBIT_CC)] <- "U" #Majority
subdat3$MAJOR_CC[is.na(subdat3$MAJOR_CC)] <- "U" #Majority
subdat3$COMPUTER_ELECTRONIC[is.na(subdat3$COMPUTER_ELECTRONIC)] <- "U" #Majority
subdat3$CHANNEL_DOMINANCE[is.na(subdat3$CHANNEL_DOMINANCE)] <- "C" #Majority

levels(subdat3$ETHNIC_MATCH)<-c(levels(subdat3$ETHNIC_MATCH),"N")
subdat3$ETHNIC_MATCH[is.na(subdat3$ETHNIC_MATCH)] <- "N" #Blanks = No

subdat3$HOMEOWNR[is.na(subdat3$HOMEOWNR)] <- "Y" #Majority
subdat3$ADD_TYPE[is.na(subdat3$ADD_TYPE)] <- "S" #Majority
subdat3$IND_DMR[is.na(subdat3$IND_DMR)] <- "U" #Majority
subdat3$MAILPREF[is.na(subdat3$MAILPREF)] <- "N" #Majority

subdat3$ZONLINE[is.na(subdat3$ZONLINE)] <- "Y" #Majority
subdat3$ZMOB[is.na(subdat3$ZMOB)] <- "Y" #Majority
subdat3$ZPRCHPHN[is.na(subdat3$ZPRCHPHN)] <- "U" #Majority
subdat3$ZMOBMULT[is.na(subdat3$ZMOBMULT)] <- "Y" #Majority
subdat3$ZHMDECOR[is.na(subdat3$ZHMDECOR)] <- "Y" #Majority
subdat3$ZHOMEENT[is.na(subdat3$ZHOMEENT)] <- "U" #Majority
subdat3$ZKITCHEN[is.na(subdat3$ZKITCHEN)] <- "U" #Majority
subdat3$ZPRCHONL[is.na(subdat3$ZPRCHONL)] <- "Y" #Majority
subdat3$SALES[is.na(subdat3$SALES)] <- "U" #Unknown

subdat2$ZIP[is.na(subdat2$ZIP)] <- "60091" #Majority

#Impute Numeric Variables with Median
subdat3$EXAGE[subdat3$EXAGE=="U"] <- NA
subdat3$EXAGE[is.na(subdat3$EXAGE)] = median(subdat3$EXAGE, na.rm = TRUE)

#Impute Numeric Variables with 0
subdat3$salepercamp[is.na(subdat3$salepercamp)] <- 0
subdat3$salepercamp[is.infinite(subdat3$salepercamp)] <- 0
subdat3$salepertrans[is.na(subdat3$salepertrans)] <- 0
subdat3$NUM_CHILD[is.na(subdat3$NUM_CHILD)] <- 0

#Check for Missingness
sum(is.na(subdat3))
```

```
sapply(subdat3, function(x) sum(is.na(x)))
aggr_plot <- aggr(subdat3, col=c('#9ecae1','#de2d26'), numbers=TRUE,prop=FALSE, so
rtVars=TRUE, labels=names(subdat3), cex.axis=.5, gap=2, ylab=c("Histogram of missi
ng data","Pattern"))

#Check missing data percentage
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(subdat3,2,pMiss)

str(subdat3)

######################################################################
###########  Random Forest model for XYZ Didn't Target#############
######################################################################

###Random Forest Model###
library(caret)
detach(package:ModelMetrics)

prob <- predict(rf.model, newdata = subdat3, type = "prob")[,2]
hist(prob)

#Confusion Matrix for Probabilities
library(caret)

glm.pred  <- ifelse(prob  > 0.50,1,0)
xtab.RF1=table(glm.pred, subdat3$RESPONSE16)
confusionMatrix(xtab.RF1, positive = "1") #Accuracy: 0.7925;  Sensitivity : 0.4253
5;  Specificity : 0.82129

#Confusion Matrix Detail
confusion.matrix <- table(prob, subdat3$RESPONSE16)
confusion.matrix

#Plot Curves
require(PRROC)
set.seed(1)

fg <- prob[subdat3$RESPONSE16 == 1]
bg <- prob[subdat3$RESPONSE16 == 0]

#ROC Curve
roc <- roc.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
plot(roc)

#Precision Recall Curve
pr <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
plot(pr)

#Compute area under ROC curve
roc <- roc.curve( fg, bg );
print( roc ) #0.67757

#Compute area under PR curve
pr <- pr.curve( fg, bg )
print( pr ) #0.1790455

#AUC
#library(ModelMetrics)
```

```
#prob <- predict(rf.model, newdata=subdat3, type='prob')
#auc<-auc(subdat3$RESPONSE16,prob[,2])
#auc #0.67757

## Evaluate on All data in subset
prediction_rf_all2<- predict(rf.model, newdata = subdat3, type = "prob")[,2]

pred2df2 <- as.data.frame(prediction_rf_all2)
data_alldf2 <- cbind(subdat3,prediction_rf_all2)
str(data_alldf2)

#Average Sales Per Transactions
mean(data_alldf2$salepertrans) #Average subset is $204.4749
mean(data_alldf2$salepertrans[data_alldf2$prediction_rf_all2>0.70]) #70% probabili
ty is 212.1587

#Average Revenue per Customer for All
mean(data_alldf2$REVPERCUST) #Average subset is $118.8639

#Targeted Customers with New Model
sum(data_alldf2$prediction_rf_all2>0.65) #1387
sum(data_alldf2$prediction_rf_all2>0.5) #3118
sum(data_alldf2$prediction_rf_all2>0.4) #5002
sum(data_alldf2$prediction_rf_all2>0.3) #7931
sum(data_alldf2$prediction_rf_all2>0.2) #11536
sum(data_alldf2$prediction_rf_all2>0.15) #13335
sum(data_alldf2$prediction_rf_all2>0.10) #14717
sum(data_alldf2$prediction_rf_all2>0.05) #15598

#Average Revenue per Customer for New Model
mean(data_alldf2$REVPERCUST[data_alldf2$prediction_rf_all2>0.65]) #466.9753
mean(data_alldf2$REVPERCUST[data_alldf2$prediction_rf_all2>0.5]) #339.6213
mean(data_alldf2$REVPERCUST[data_alldf2$prediction_rf_all2>0.4]) #263.3613
mean(data_alldf2$REVPERCUST[data_alldf2$prediction_rf_all2>0.3]) #191.4004
mean(data_alldf2$REVPERCUST[data_alldf2$prediction_rf_all2>0.2]) #143.3182
mean(data_alldf2$REVPERCUST[data_alldf2$prediction_rf_all2>0.15]) #129.8518
mean(data_alldf2$REVPERCUST[data_alldf2$prediction_rf_all2>0.10]) #122.6703
mean(data_alldf2$REVPERCUST[data_alldf2$prediction_rf_all2>0.05]) #119.5404

str(prediction_rf_all2)
```