KDD Cup 2009 - Customer Relationship Prediction

Udy Akpan, Joe Dion, Sandra Duenas, Manjari Srivastava, Jay Swinney

July 10, 2015

Contents

1 Introduction

The KDD Cup 2009 CRM problem is a prediction problem attempting to predict account cancellation (churn), account purchase of additional services (appetency), and account willingness to respond favorably to marketing pitches (up-selling).
The prediction of each response (churn, appetency, up selling) is performed separately using classification.

Given that the predictors are unknown and they are generically labeled, data imputation of missing values is done by using zero (0) for numeric variables and 'missing' for categorical variables. An indicator variable for fixed or imputed values is set to 1 to identify those observations and variable combination that was imputed.

Naive models were used to perform initial EDA and helped in model specification by determining the best set of variables to be included in the initial models created with only the training data.

2 The Modeling Problem

The problem requires the prediction of three (3) different response variables, churn, appetency, and up selling. These variables are not multinomial or different values in a categorical variable. These are independent response variables.

The nature of the response variable dictates as to whether an account churns or not, or whether it up-sells or not, or whether it appetences or not calls for a classification approach to the prediction problem.

The data set contain 230 variables of which nothing is known. The variable names are generic. There are many missing values but due to the unknown nature of the data, imputation may be narrowed to the variable itself.

3 The Data

4 Exploratory Data Analysis

4.1 Imputing Missing Data

Our strategy to impute missing data is to replace missing numeric values with a 0 and then create a boolean variable that indicates missingness. For categorical variables, all classes that represent less than 1% of the total observations were grouped into an "other" category, then a separate missing class was created.The categorical variables were imputed with the word 'missing'. The new missing indicator variables were set to 1 to indicate that the variable was imputed or to 0 to indicate no imputation.

Create testing and training data sets as well as a matrix form of the data that is required by some of the classifiers used in this analysis.

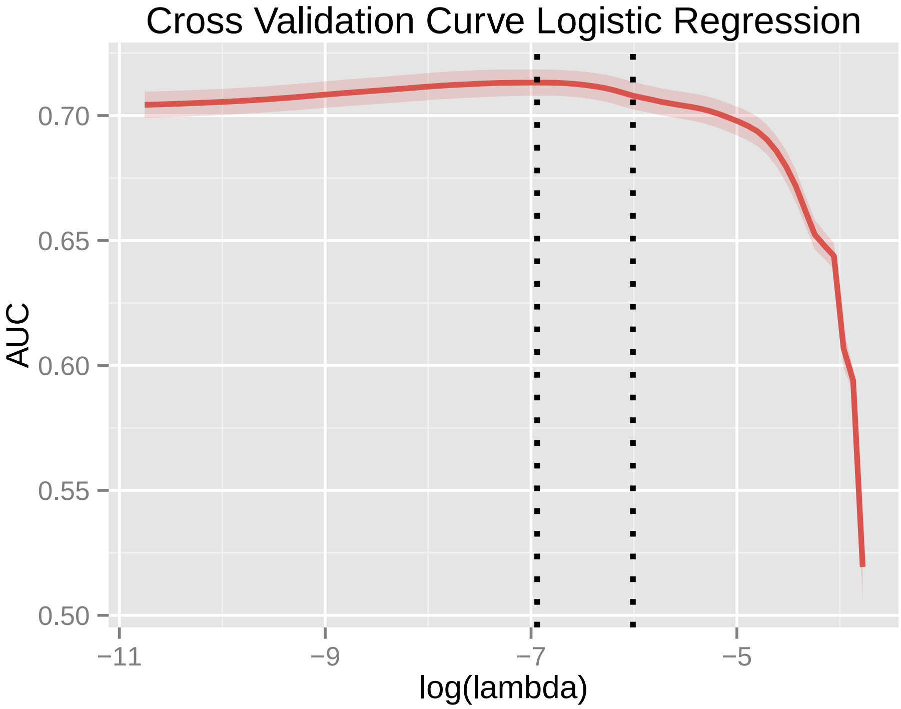
4.2 Churn

The challenge for the KKD cup 2009 consisted of predicting 3 variables from the same data set. This paper will focus on one variable at a time starting with churn.

4.2.1 Logistic Regression with Elastic-Net Penalty

A useful technique for understanding which variables have predictive power is to apply logistic regression with a regularization term. In this case elastic-net penalty is used to explore the predictive importance of the variables.

http://www.stanford.edu/~hastie/glmnet/glmnet_alpha.html



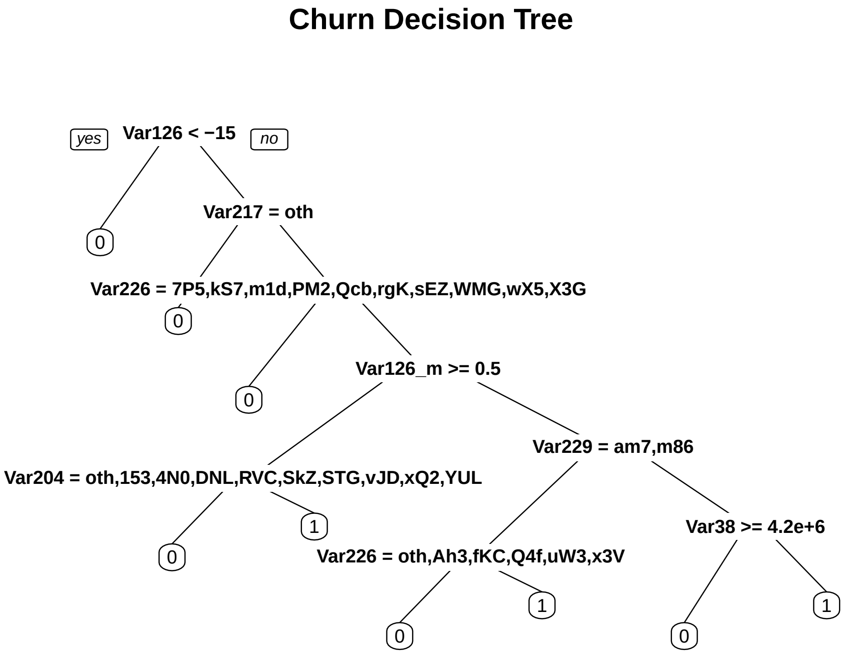
This plot shows that not all the variables are useful for classification. Two vertical lines in this plot represent the model with the best performance and the most regularized model within one standard deviation of the top performer. Performance is measured on out of sample data. The regularized and cross validated logistic regression selected a model with 155 non-zero variables

Some of the variables selected by the regularized logistic regression are in the table below with their coefficients. Only selected variables are shown for brevity.

Table 1: Variables Selected by Elastic-Net

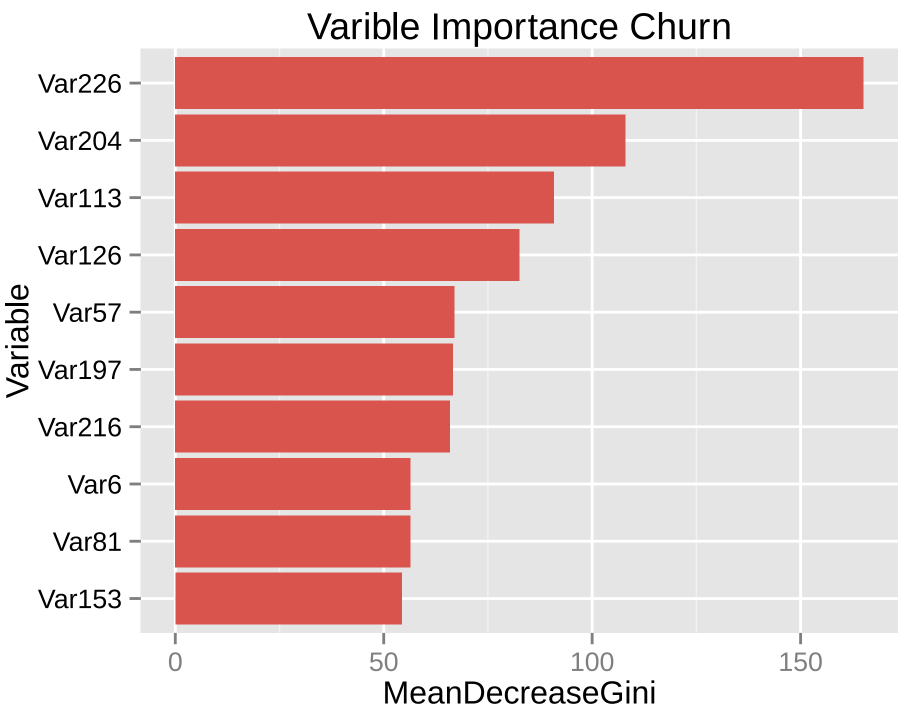
variable	coeficient
Var126	0.2553552
Var126_missing	0.3573078
Var226_dummy_FSa2	0.0643113
Var226_dummy_PM2D	0.0319844
Var226_dummy_me1d	-0.4173382
Var226_dummy_TNEC	0.1534265
Var226_dummy_uWr3	0.0840565
Var226_dummy_7P5s	-0.0186982

4.2.2 Decision Tree

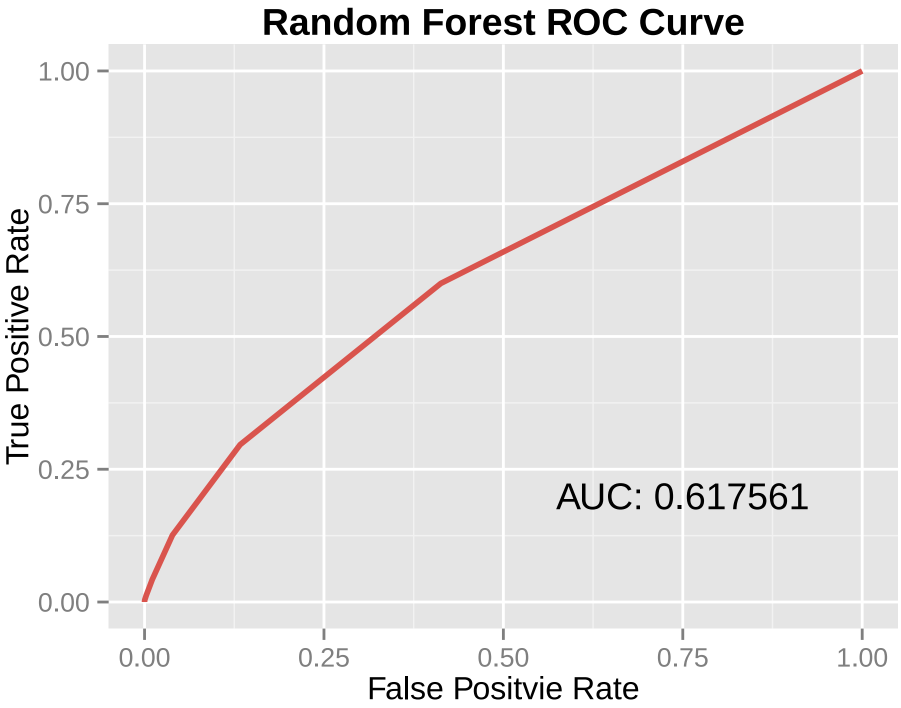

Churn Decision Tree
yes
Var126 < −15
no
0
Var217 = oth
Var226 = 7P5,kS7,m1d,PM2,Qcb,rgK,sEZ,WMG,wX5,X3G
0
0
Var126_m >= 0.5
Var229 = am7,m86
Var204 = oth,153,4N0,DNL,RVC,SkZ,STG,vJD,xQ2,YUL
1
0
Var38 >= 4.2e+6
Var226 = oth,Ah3,fKC,Q4f,uW3,x3V
1
1
0
0

The results from the decision tree give an indication of how useful each of the variables are at predicting churn. This tree is fairly shallow, so any of the variables that made it into the tree will most likely show up in other models that give some indication of variable importance. One interesting thing to not about this tree is that variables 126 and 226 both show up twice in the tree, confirming what has been seen from the logistic regression with elastic net penalty and the random forest variable importance in the next section.

4.2.3 Random Forest



With the random forest as with the decision tree and logistic regression Var226 has shown to be an important indicator of churn. Variable 204 also shows up high in the variable importance plot from the random forest and in the single decision tree from the previous section.

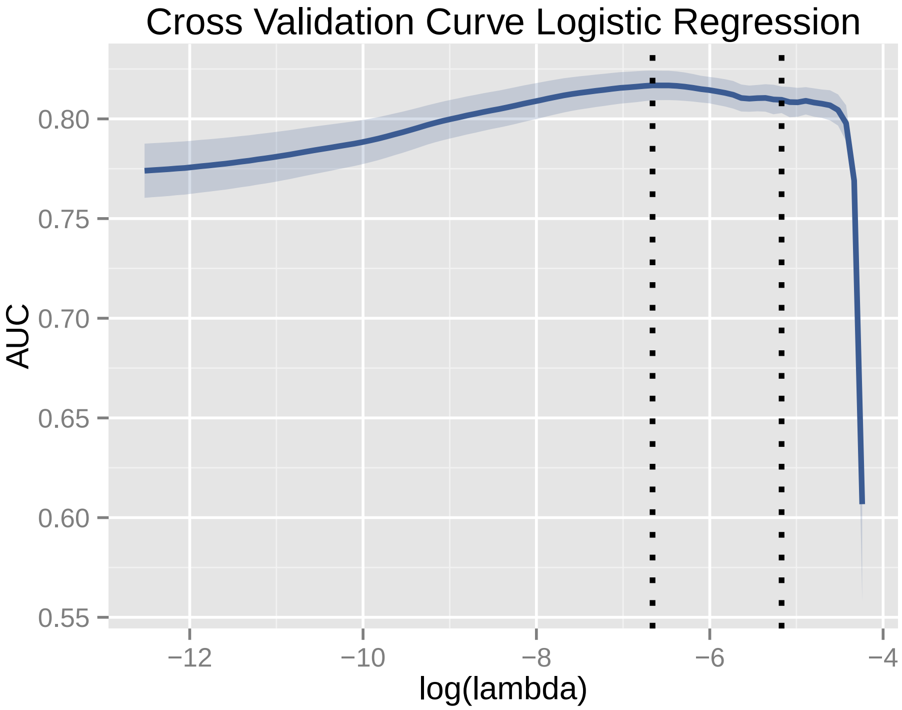


The accuracy of the random forest leaves something to be desired, there is clearly more work to do. It is not displayed here, but the random forest fit extremely well to in-sample data, this indicates that there is more work to be done to combat over-fitting. Options include changing the requirements for leaf and split sizes and trying the random forest with a subset of variables such as the ones selected by regularized logistic regression.

4.3 Appetency

The next response variable to discuss is appetency. As defined in the task description on the KDD website, appetency is the propensity to buy a service or a product.

4.3.1 Logistic Regression with Elastic-Net Penalty

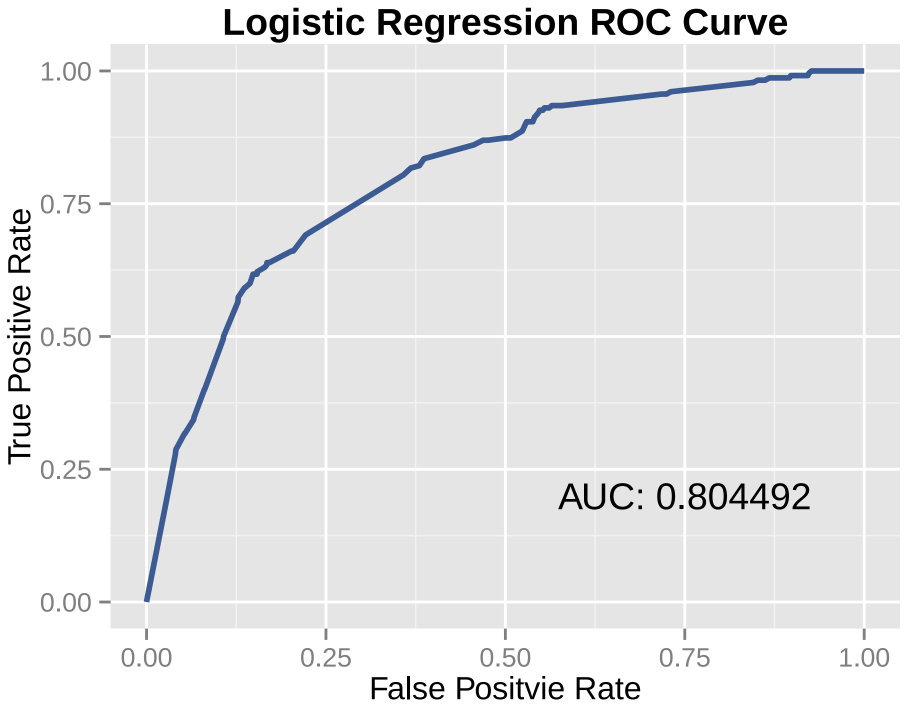


The results from the logistic regression are very promising and interesting. The AUC peaks above 0.8 which is nice to see, but more interestingly the AUC does not dramatically decline till almost all of the variables are removed from the model. This shows that one or two of the variables are very strong indicators of appetency.

Taking a look at the 3 variables in the highly regularized model (right-most vertical line) shows that Var126 and a certain level of Var218 plus an intercept are very indicative of appetency. This is encouraging because it suggest that predicting appetency will be an easier problem.

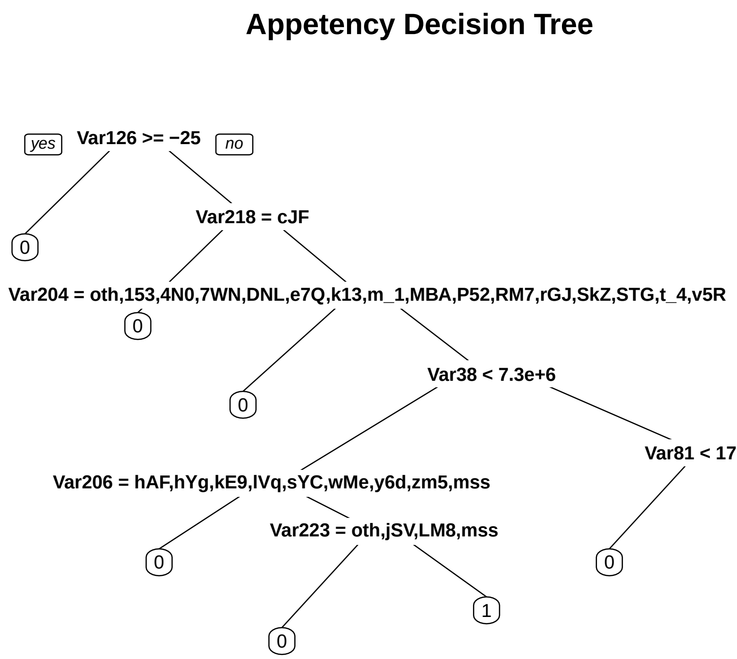
Table 2: Variables Selected by Elastic-Net

	coeficient
(Intercept)	-3.9459138
Var126	-0.5846417
Var218_dummy_cJvF	-0.7195445
Var218_dummy_UYBR	0.1148212



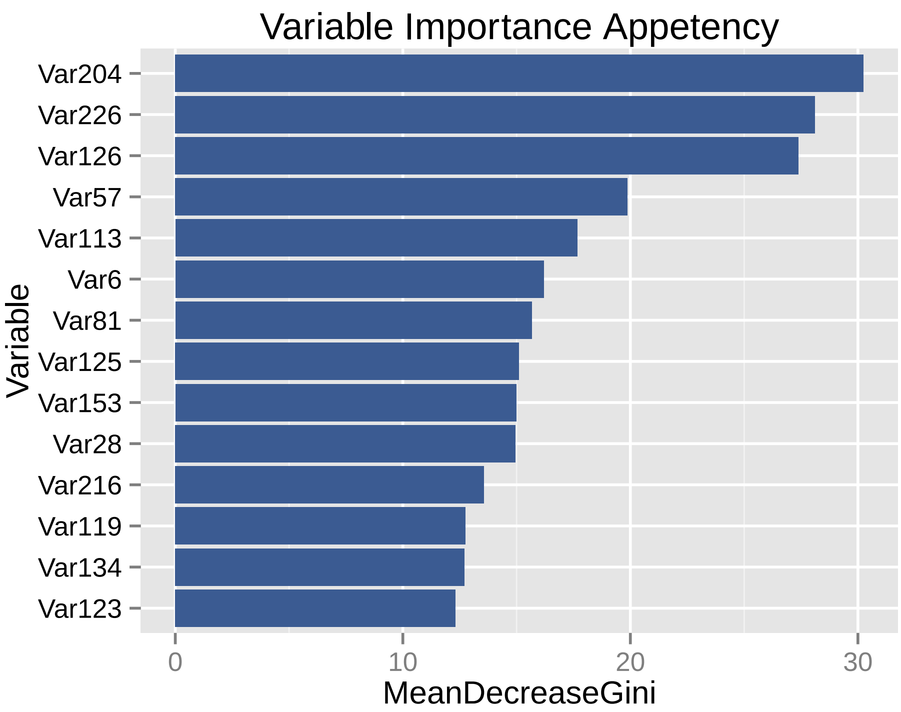
As shown by this ROC curve constructed on out of sample data, the logistic regression performs very well identifying appetency.

4.3.2 Decision Tree

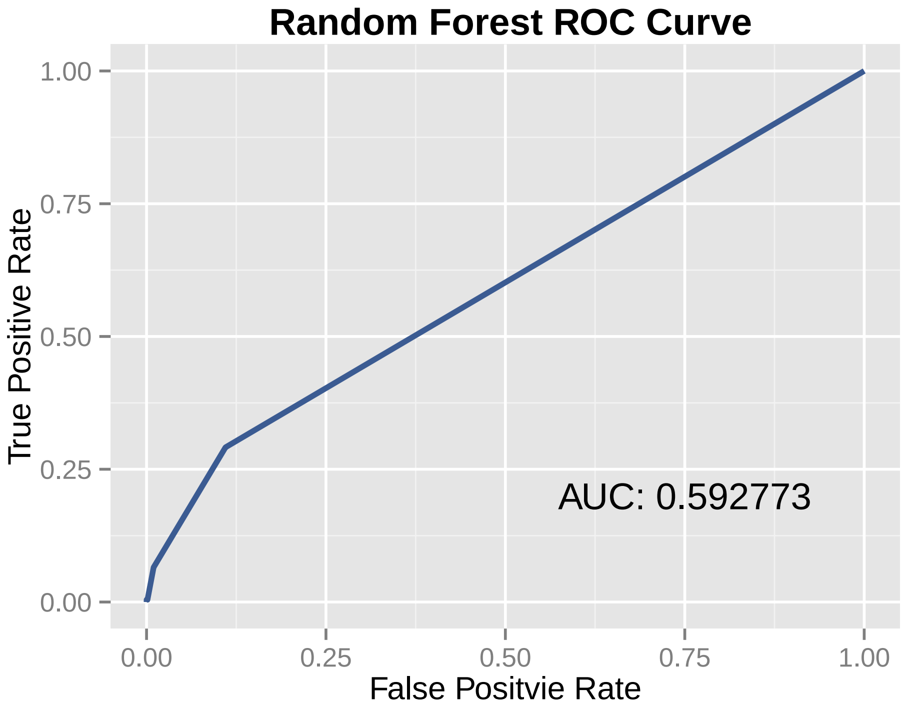

Appetency Decision Tree
yes
Var126 >= −25
no
0
Var218 = cJF
Var204 = oth,153,4N0,7WN,DNL,e7Q,k13,m_1,MBA,P52,RM7,rGJ,SkZ,STG,t_4,v5R
0
0
Var38 < 7.3e+6
Var81 < 176e+3
Var206 = hAF,hYg,kE9,lVq,sYC,wMe,y6d,zm5,mss
Var223 = oth,jSV,LM8,mss
1
0
0
1
0

The Decision Tree classifier selected 7 variables as the most predictive variables. The 7 variables are listed below with the highest predictive value listed first.
The following configuration: minsplit=40 to set the minimum number of observations per node, minbucket=10 to set the minimum number of total nodes, and cp=0.001 to set the cost complexity factor with a split that must decrease the overall lack of fit by a factor of 0.001. 1. Var126
2. Var218
3. Var204
4. Var38
5. Var206
6. Var223
7. Var81

4.3.3 Random Forest



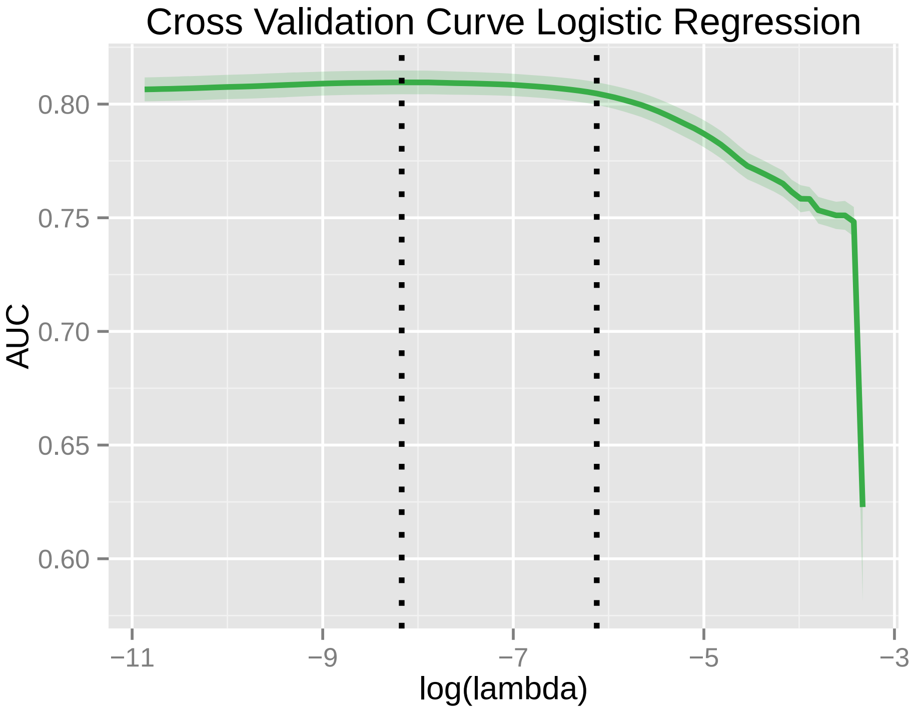
An interesting take away from this plot is that the random forest identified variable 226 & 126 as two of the top three most important variables. This echos the output from the logistic regression and further confirms that these are important variables. However it will be shown that the random forest did not perform nearly as well as the regularized logistic regression, this is because the random forest is severely over-fit, it will need significant tuning before it is on par with the regularized logistic regression.



4.4 Up-Sell

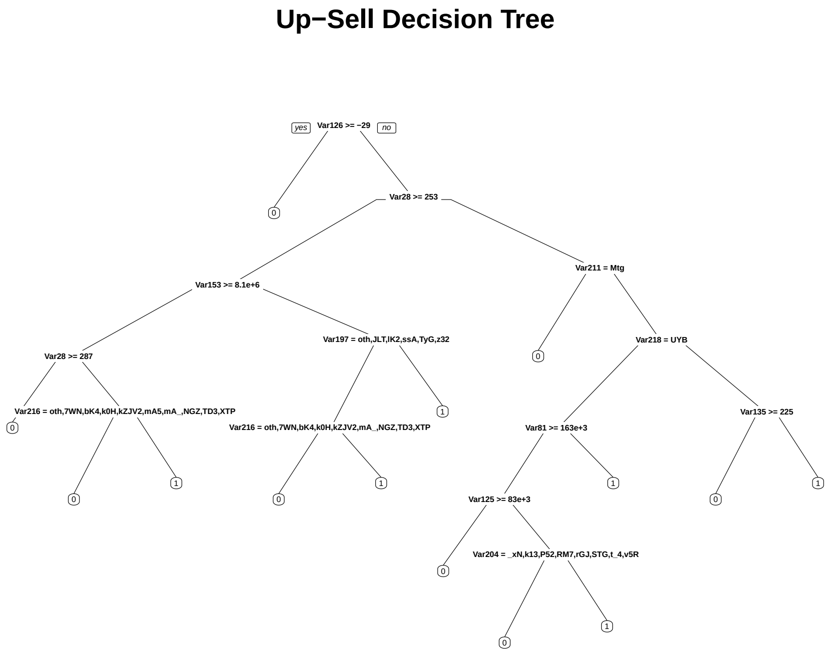
The last response variable to analyze is up-sell. As defined in the task description, up-selling can imply selling something additional, or selling something that is more profitable or otherwise preferable for the seller instead of the original sale.

4.4.1 Logistic Regression with Elastic-Net Penalty



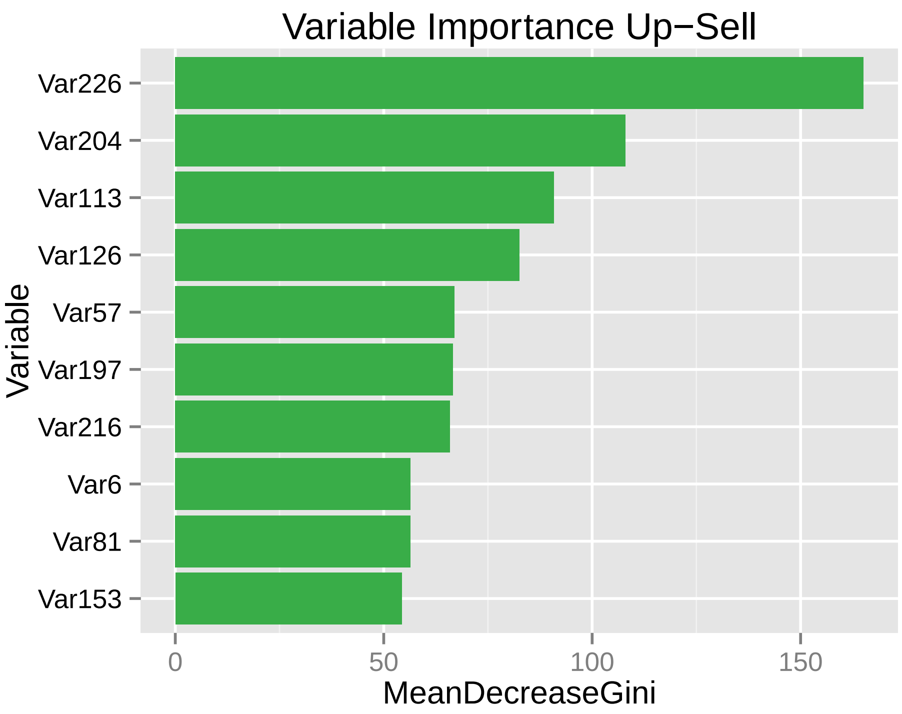
The results from the regularized logistic regression are both promising and somewhat disappointing at the same time. The plot shows that comparable performance can be achieved by removing all but about 80 variables, but unfortunately the regularization does not appear to yield much in the way of performance gain.

4.4.2 Decision Tree


Up−Sell Decision Tree
yes
Var126 >= −29
no
0
Var28 >= 253
Var153 >= 8.1e+6
Var211 = Mtg
Var28 >= 287
Var197 = oth,JLT,lK2,ssA,TyG,z32
0
Var218 = UYB
Var216 = oth,7WN,bK4,k0H,kZJV2,mA5,mA_,NGZ,TD3,XTP
0
1
Var216 = oth,7WN,bK4,k0H,kZJV2,mA_,NGZ,TD3,XTP
Var81 >= 163e+3
Var135 >= 225
0
1
0
1
1
0
1
Var125 >= 83e+3
0
Var204 = _xN,k13,P52,RM7,rGJ,STG,t_4,v5R
0
1

A number of control options were used for rpart() -, namely minsplit to set the minimum number of observations per node, minbucket - minimum number of total nodes , cp - split must decrease the overall lack of fit by a factor of 0.001 (cost complexity factor). It was observed that Var 126 and Var 28 were chosen and always had a high importance.These variables probably have good predictive value for up-sell.

4.4.3 Random Forest



The Random Forest classifier selects close to 200 predictor variables as having significant predictive value for up-sell. This is a very large number. We do see that the mean decrease in Gini Index is highest on including Var 126. This matches with the results from Decision tree that Var 126 has a higher predictor value.

4.5 Other Analysis

In addition to the steps listed here, KNN and PCA was used to explore all of the response variables without any interesting results. Before any model fitting was done, each predictor variable was examined individually to look for patterns and structure manually. The uni-variate work is not included here because it is too space consumptive.

5 Predictive Modeling: Methods and Results

5.1 Train/Test Data

5.2 Appetency

5.2.1 Naïve Bayes

The Naïve Bayes technique was applied in a computational EDA manner to obtain the highest AUC score for appetency.

The variable selection process was based on the smallest deviance of each variable. This variable selection process resulted in 31 variables out of 230 with deviance of 186.294 based on the Calibration data set.

The Calibration data set is a 10% random selection of observations from the original data set.

The resulting Naive Bayes model using the selected variables shows that the model is overfitting the data because the AUC Score with the Train data is 0.9619 but the AUC Score with the Test data is 0.7624, which is about a 20-point difference. However, the AUC for the Test is significantly above 0.50 of a random guess, so we could consider the Naive Bayes model for appetency to be reasonably accurate.

5.2.2 Random Forest

The Random Forest classifier was used to build a classification model for Appetency variable on the training data set. The parameters chosen were number of trees = 50 and minimum bucket size =10 . The first model was built choosing all the 230 variables and the imputed variables. This resulted in a random forest model with over 200 variables. This model however was not able to detect appetency=1 cases in the test data set in spite of a decent accuracy level of 98.6% in test data set.

Multiple models were created using a subset of variables based on importance - top 25 and top 50 variables. These models did have a high accuracy but were not able to detect appetency=1 cases in test data set. Also models were created using the sample size option (sampsize = c(10,30)) , this implies that the algorithm will randomly draw 10 and 30 from two values of appetency = 0 and 1 to grow the tree. This improved the accuracy of the model, but appetency =1 customers were still not predicted correctly. Random forest may not be a good classifier for appetency.

Area under the curve (AUC) is used to determine goodness of fit for the model. AUC for Random forest model for Appetency in test data is 0.622. This is a low value, so the model is only slightly better than a random pick.

5.2.3 Logistic Regression

5.2.4 K-Nearest Neighbors

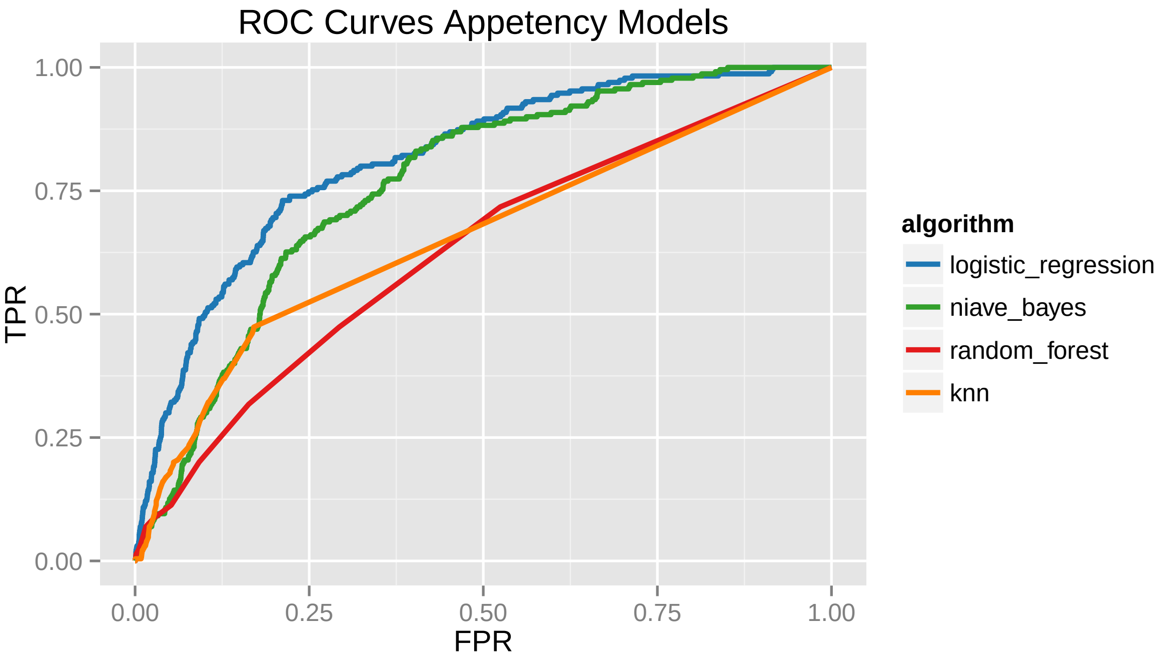
The Nearest Neighbor K technique was applied in a computational EDA manner to obtain the highest AUC score for appetency.

The variable selection process was based on the smallest deviance of each variable. This variable selection process resulted in 31 variables out of 230 with deviance of 504.483 based on the Calibration data set.

The Calibration data set is a 10% random selection of observations from the original data set.

The resulting knn model used the selected variables and k = 200. It shows that the model is overfitting the data because the AUC Score with the Train data is 0.9904 but the AUC Score with the Test data is 0.6548, which is about a 33-point difference. The AUC for the Test is somewhat significantly above 0.50 of a random guess, so we could consider the knn model for upsell to be reasonably accurate.

5.2.5 Model Performance



5.3 Churn

5.3.1 Naïve Bayes

The Naïve Bayes technique was applied in a computational EDA manner to obtain the highest AUC score for churn.

The variable selection process was based on the smallest deviance of each variable. This variable selection process resulted in 47 variables out of 230 with deviance of 291.862 based on the Calibration data set.

The Calibration data set is a 10% random selection of observations from the original data set.

The resulting Naive Bayes model using the selected variables shows that the model is overfitting the data because the AUC Score with the Train data is 0.9315 but the AUC Score with the Test data is 0.6622, which is about a 27-point difference. However, the AUC for the Test is significantly above 0.50 of a random guess, so we could consider the Naive Bayes model for churn to be reasonably accurate.

5.3.2 Random Forest

The Random Forest classifier was used to build a classification model for Churn variable on the training data set. The parameters chosen were, number of trees = 50 and minimum bucket size =10 . The first model was built choosing all the 230 variables and the imputed variables. This resulted in a random forest model with over 200 variables. This model however was not able to detect churn=1 cases in the test data set in spite of a decent accuracy level of 92.3% in test data set.

A second model was then built using the top 50 variables based on importance from the first model. This model showed higher accuracy percentage (92.4%). The model was able to detect churn=1 scenarios better than the previous random forest model. Area under the curve (AUC) is used to determine goodness of fit for the model. AUC for Random forest model for Churn in test data is 0.6883. This is a low value, so the model is better than a random pick but it is not a good model.

5.3.3 K-Nearest Neighbors

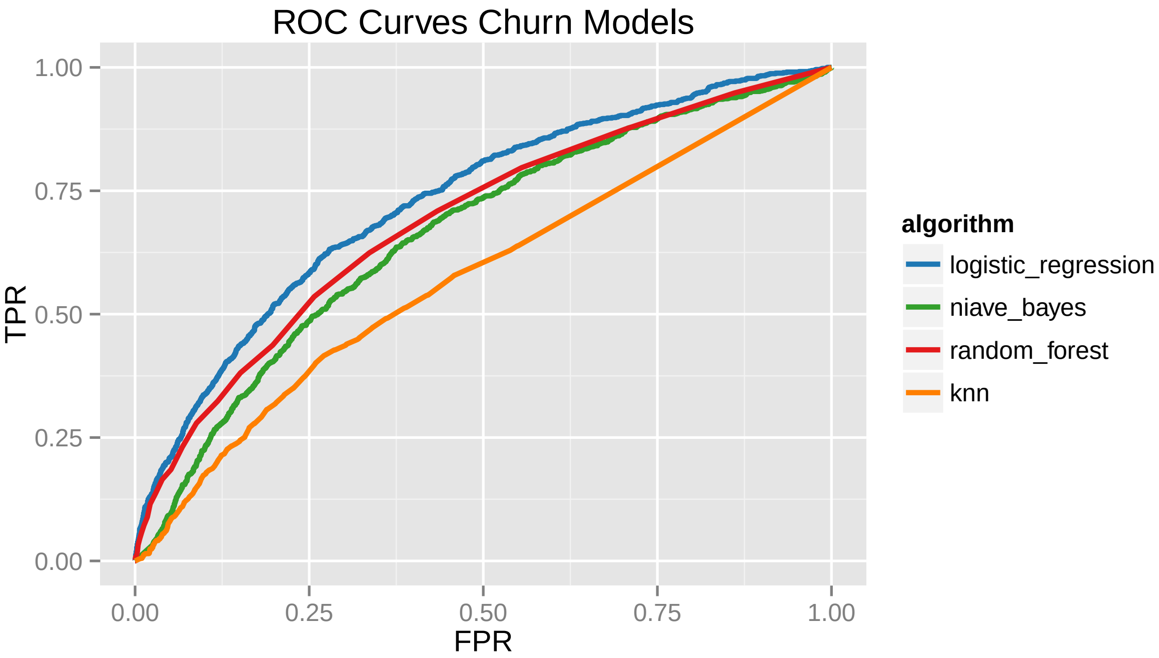
The Nearest Neighbor K technique was applied in a computational EDA manner to obtain the highestAUC score for churn.

The variable selection process was based on the smallest deviance of each variable. This variable selection process resulted in 47 variables out of 230 with deviance of 504.483 based on the Calibration data set.

The Calibration data set is a 10% random selection of observations from the original data set.

The resulting knn model used the selected variables and k = 200. It shows that the model is overfitting the data because the AUC Score with the Train data is 0.9801 but the AUC Score with the Test data is 0.5751, which is about a 30-point difference. The AUC for the Test is *not* significantly above 0.50 of a random guess, so we could not consider the knn model for upsell to be reasonably accurate.

5.3.4 Model Performance



5.4 Up-Sell

5.4.1 Naïve Bayes

The Naïve Bayes technique was applied in a computational EDA manner to obtain the highest AUC score for upsell.

The variable selection process was based on the smallest deviance of each variable. This variable selection process resulted in 51 variables out of 230 with deviance of 504.483 based on the Calibration data set.

The Calibration data set is a 10% random selection of observations from the original data set.

The resulting Naive Bayes model using the selected variables shows that the model is overfitting the data because the AUC Score with the Train data is 0.9177 but the AUC Score with the Test data is 0.7515, which is about a 16-point difference. However, the AUC for the Test is significantly above 0.50 of a random guess, so we could consider the Naive Bayes model for upsell to be reasonably accurate.

5.4.2 Random Forest

The Random Forest classifier was used to build a classification model for Upsell variable on the training data set. The parameters chosen were number of trees = 50 and minimum bucket size =10 . The first model was built choosing all the 230 variables and the imputed variables. This resulted in a random forest model with over 200 variables. This model however was not able to detect upsell=1 cases in the test data set in spite of a decent accuracy level of 98.6% in test data set. A second model was then built using the top 25 variables based on importance from the first model. This model showed higher accuracy percentage (95.14%) in test data set. The model was able to detect upsell=1 scenarios better than the previous random forest model.

Area under the curve (AUC) is used to determine goodness of fit for the model. AUC for Random forest model for Upsell in test data is 0.8334. This is a moderate value, so the model is not a perfect fit.

5.4.3 Logistic Regression

5.4.4 K-Nearest Neighbors

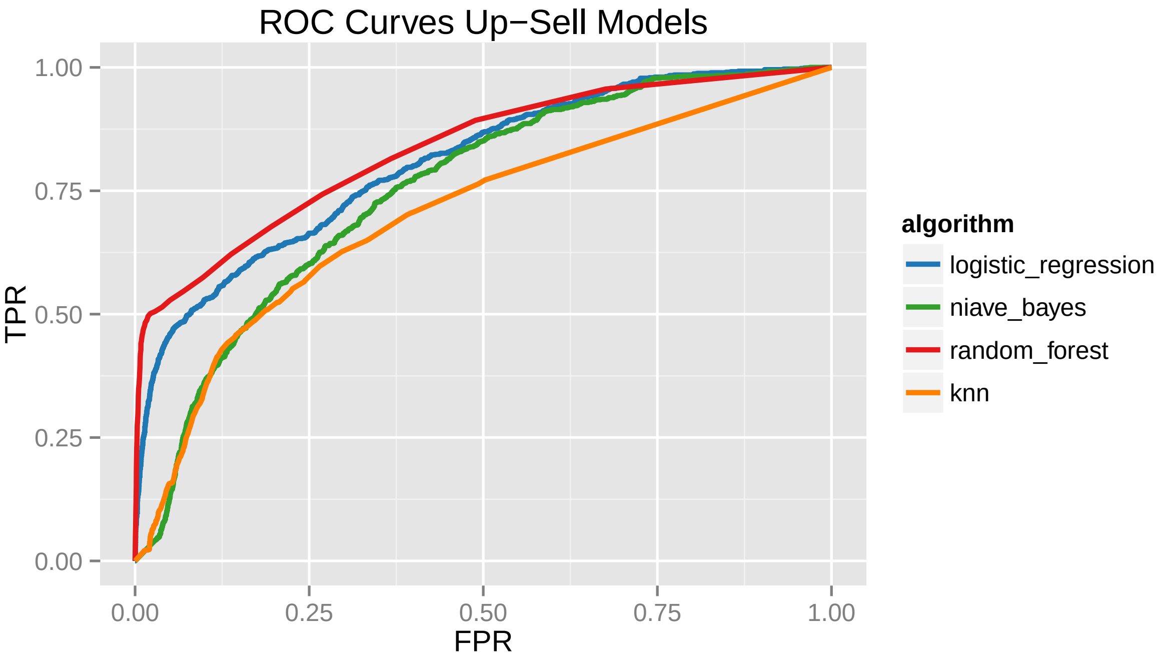
The Nearest Neighbor K technique was applied in a computational EDA manner to obtain the highest AUC score for upsell.

The variable selection process was based on the smallest deviance of each variable. This variable selection process resulted in 51 variables out of 230 with deviance of 504.483 based on the Calibration data set.

The Calibration data set is a 10% random selection of observations from the original data set.

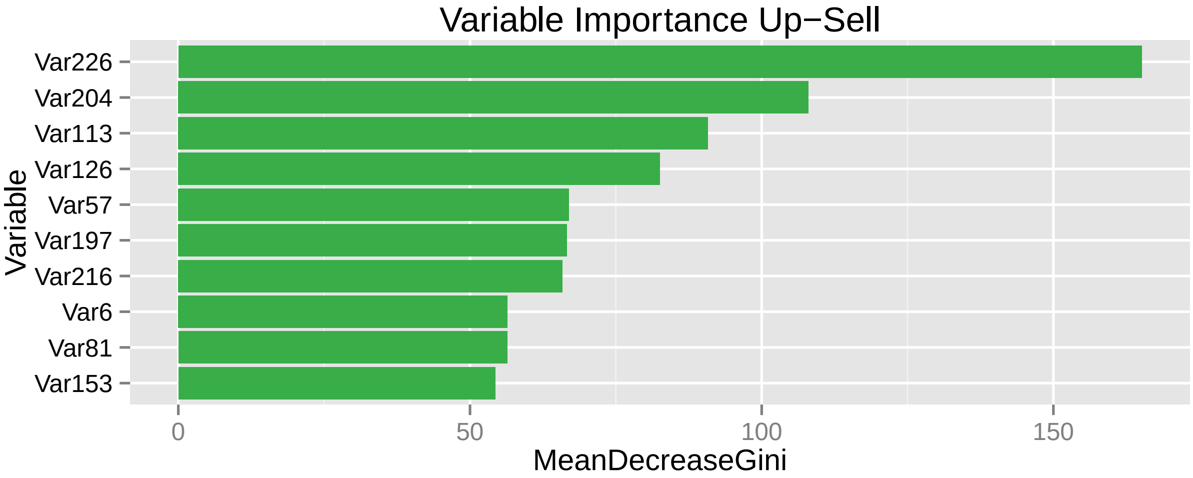
The resulting knn model used the selected variables and k = 200. It shows that the model is overfitting the data because the AUC Score with the Train data is 0.9878 but the AUC Score with the Test data is 0.7021, which is about a 20-point difference. However, the AUC for the Test is significantly above 0.50 of a random guess, so we could consider the knn model for upsell to be reasonably accurate.

5.4.5 Model Performance



6 Comparison of Results

7 Conclusions


Variable Importance Up−Sell
Variable
Var226
Var204
Var113
Var126
Var57
Var197
Var216
Var6
Var81
Var153
0
50
100
150
MeanDecreaseGini