KDD Cup 2009 - Customer Relationship Prediction

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

Summer Quarter 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. This was done for some but not all of the analysis, techniques like Naïve Bayes did not require creating 'missing' variables.

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 challenge consists of several classification problems. The goal is to make the best possible predictions of a binary target variable from a number of predictive variables. Part of the challenge deals with automatic data preparation and modeling in the context of industrial real data. Filtering constant data is the easy part of the challenge.

Another aspect of the challenge is to beat the in-house system developed by Orange Labs. It is an opportunity to prove that you can deal with a very large database, including heterogeneous noisy data (numerical and categorical variables), and unbalanced class distributions. The Orange Labs platform implements several processing methods for instances and variables selection, prediction and indexation based on an efficient model combined with variable selection regularization and model averaging method. The main characteristic of this platform is its ability to scale on very large data sets with hundreds of thousands of instances and thousands of variables. The rapid and robust detection of the variables that have most contributed to the output prediction can be a key factor in a marketing application.

Since our work is based on the Small data set of 230 variables, then we need to compare our Score to the Winner of the Slow Track, which are:

First Place: University of Melbourne University of Melbourne entry: The generally satisfactory model

Churn	Appetency	Upselling	Score
0.7570	0.8836	0.9048	0.8484

Table 1: University of Melbourne

First Runner Up: Financial Engineering Group, Inc. Japan Stochastic Gradient Boosting

Churn	Appetency	Upselling	Score
0.7589	0.8768	0.9074	0.8477

Table 2: Financial Engineering Group, Inc. Japan

Second Runner Up: National Taiwan University, Computer Science and Information Engineering Fast Scoring on a Large Database using regularized maximum entropy model, categorical/numerical balanced AdaBoost and selective Naive Bayes

However, the IBM Research Submission does not appear as a Winner of the Slow Track, it has the submission Score as follows

Churn	Appetency	Upselling	Score
0.7558	0.8789	0.9036	0.8461

Table 3: National Taiwan University

Churn	Appetency	Upselling	Score
0.7651	0.8819	0.9092	0.8521

Table 4: IBM

Evaluation: The performances are evaluated according to the arithmetic mean of the AUC for the three tasks (churn, appetency. and up-selling). This is what we call “Score”. Larger numerical values indicating higher confidence in positive class membership. The task for the competition is to beat the score of the in house model.

Churn	Appetency	Upselling	Score
0.7435	0.8522	0.8975	0.8311

Table 5: In House Models

3 The Data

The problem requires the prediction of three (3) different response variables, churn, appetency, and up-sell, which are independent binary responses.

The small original data set contains 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. The data has been scrambled for anonymity so nothing can be inferred from the values.

It can be clearly seen in Table 6 that the data set has a higher number of Accounts that did not cancel their contract because they have 0 in churn.

It can be clearly seen in Table 6 that the data set has a higher number of Accounts that did not purchase additional services because they have a 0 in appetency.

It can be clearly seen in Table 6 that data set has a higher number of Accounts that were not willing to respond favorably to marketing pitches because they have a 0 in up-sell.

	nobs	NAs	Minimum	Maximum	Q1	Q2	Mean	Median	Positive_Instances
churn	50000.00	0.00	0.00	1.00	0.00	0.00	0.07	0.00	3672.00
appetency	50000.00	0.00	0.00	1.00	0.00	0.00	0.02	0.00	890.00
upsell	50000.00	0.00	0.00	1.00	0.00	0.00	0.07	0.00	3682.00

Table 6: Response Variables

4 Exploratory Data Analysis

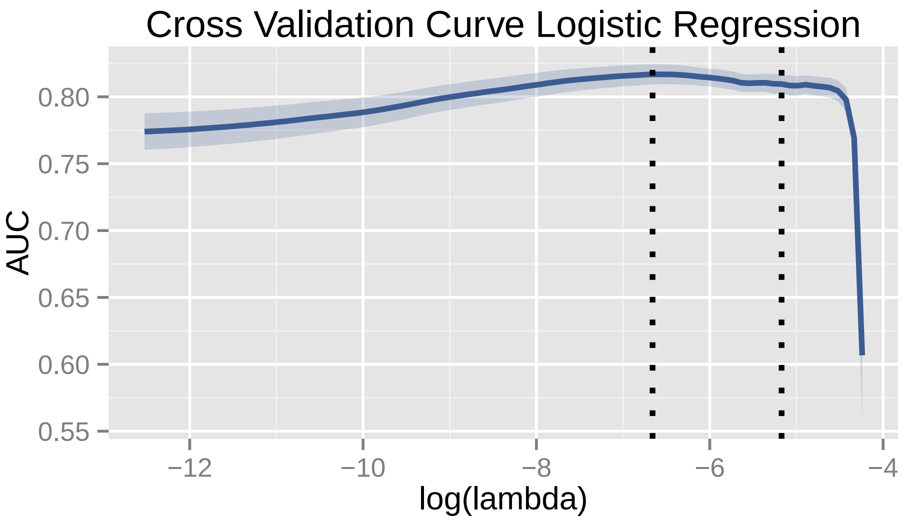
4.1 Imputing Missing Data

The most frequent strategy we used to impute missing data was to replace missing numeric values with a 0 and then create a binary 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.

4.2 Appetency

The first 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.

4.2.1 Logistic Regression with Elastic-Net Penalty

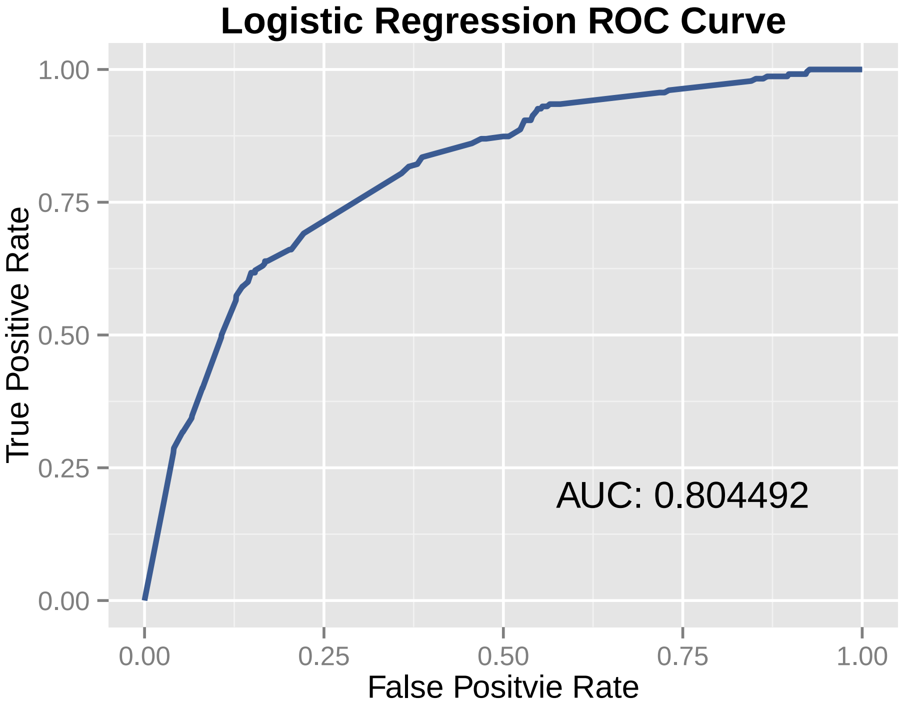


The results from the logistic regression are very promising and interesting. The AUC peaks above 0.8 which is nice to see, but more interestingly the AUC does not dramatically decline till almost all of the variables are removed from the model. This shows that one or two of the variables are very strong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

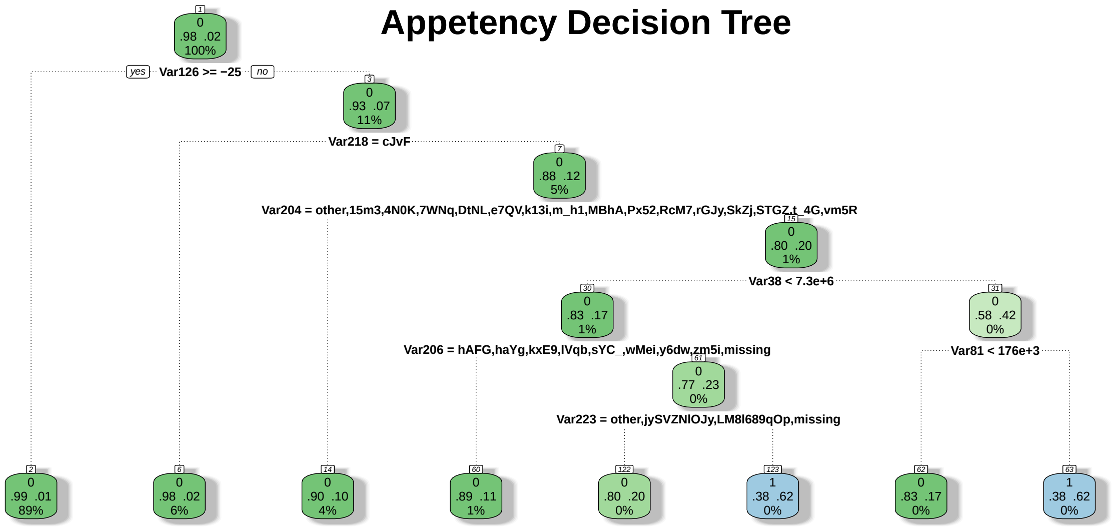
Table 7: Variables Selected by Elastic-Net

	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.

4.2.2 Decision Tree

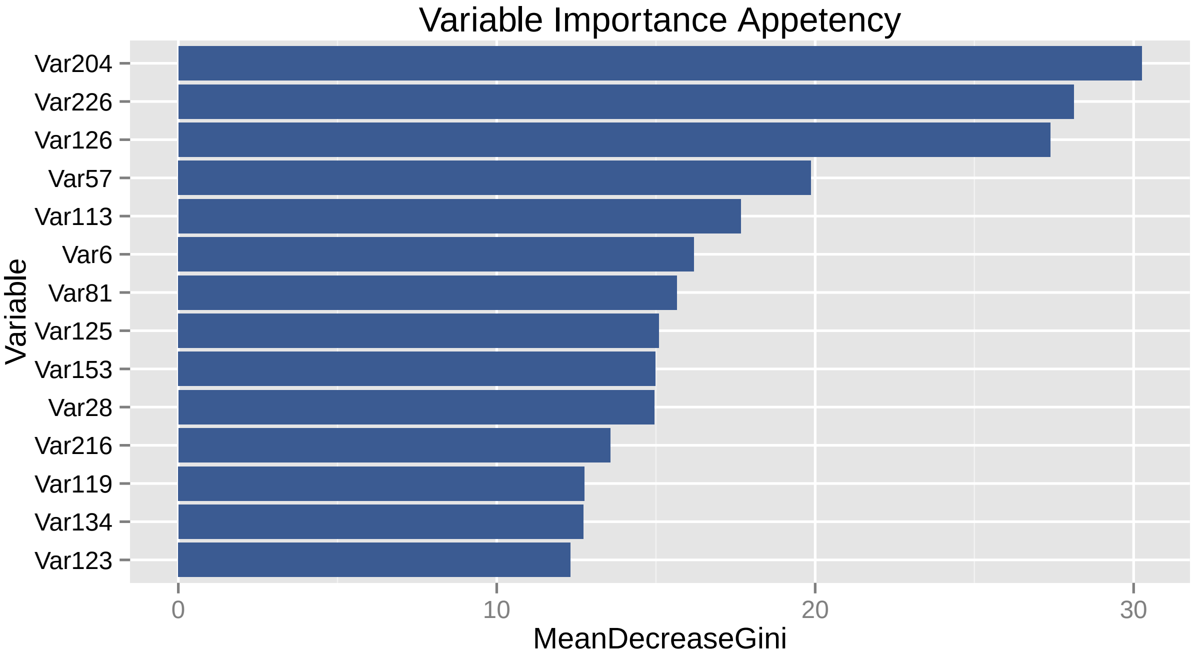


The Decision Tree classifier selected 7 variables as the most predictive variables. The 7 variables are listed below with the highest predictive value listed first.

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 5. Var206
2. Var218 6. Var223
3. Var204 7. Var81 4. Var38

4.2.3 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 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.

4.2.4 K-Nearest Neighbors & Naïve Bayes

The variable selection process for appetency 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 or less based on the Calibration data set. The Calibration data set is a 10% random selection of observations from the original data set.

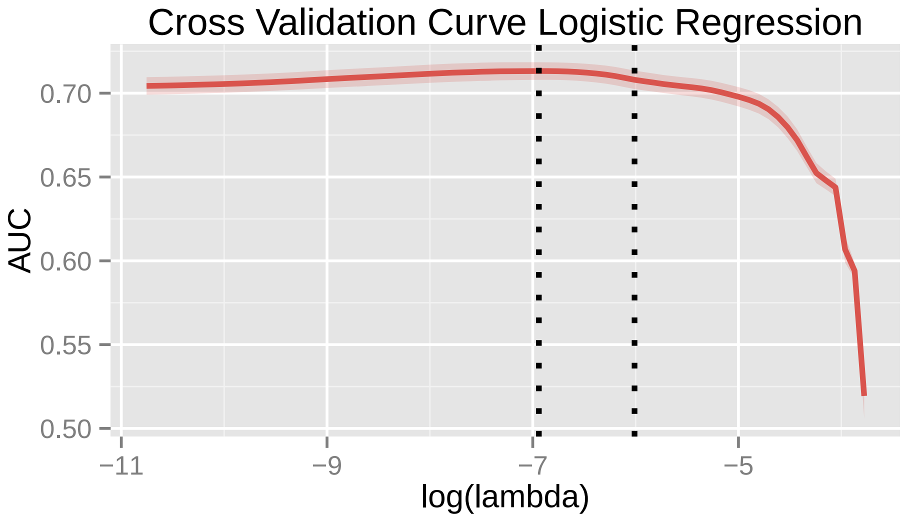
4.3 Churn

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.

4.3.1 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



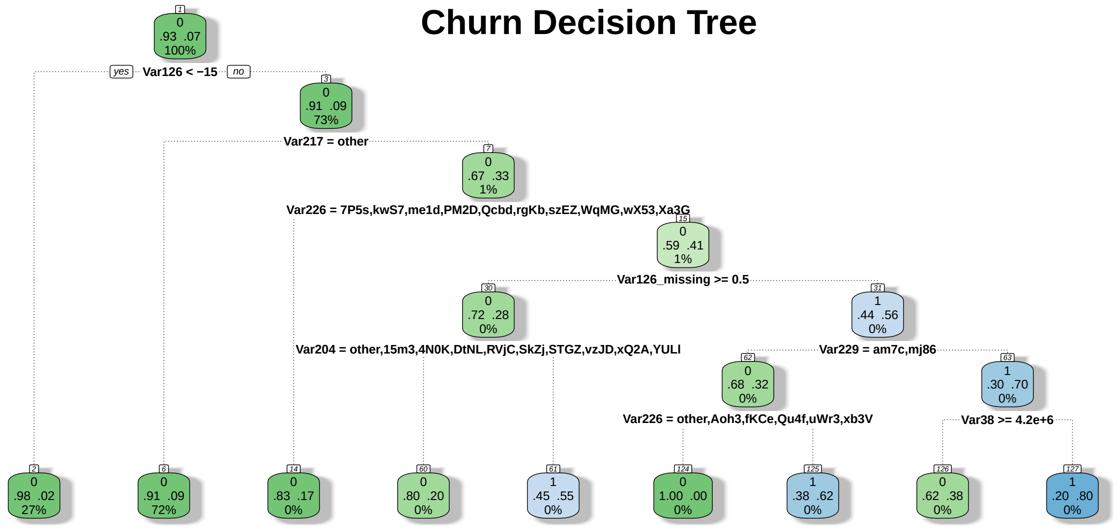
This plot shows that not all the variables are useful for classification. Two vertical lines in this plot represent the model with the best performance and the most regularized model within one standard deviation of the top performer. Performance is measured on out of sample data. The regularized and 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 8: Variables Selected by Elastic-Net

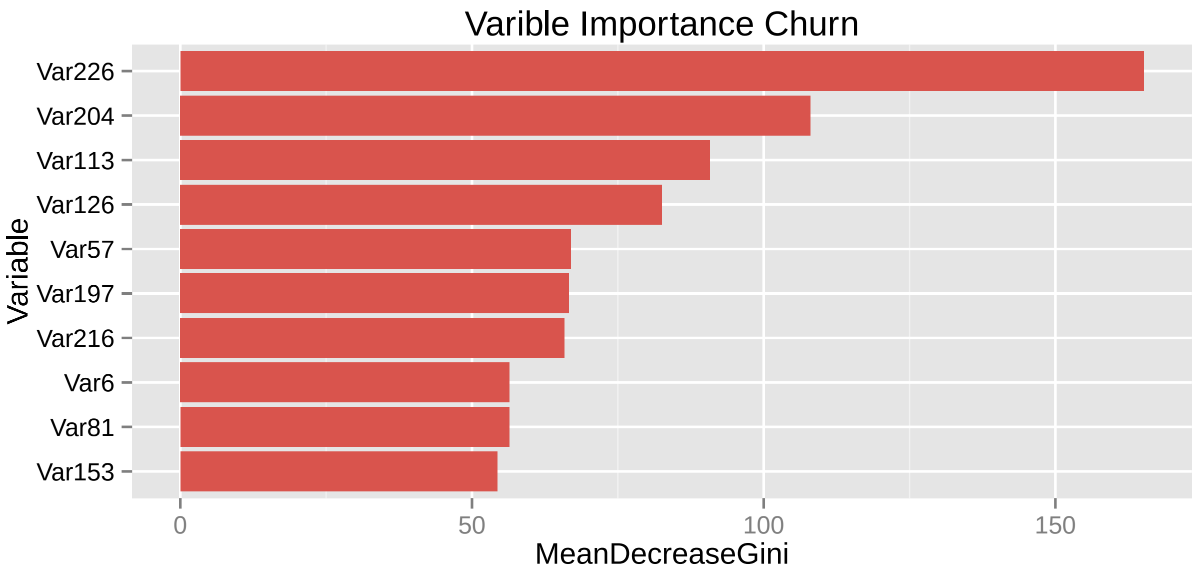
variable	coeficient
Var126	0.2553552
Var126_missing	0.3573078
Var226_dummy_FSa2	0.0643113
Var226_dummy_PM2D	0.0319844
Var226_dummy_me1d	-0.4173382
Var226_dummy_TNEC	0.1534265
Var226_dummy_uWr3	0.0840565
Var226_dummy_7P5s	-0.0186982

4.3.2 Decision Tree

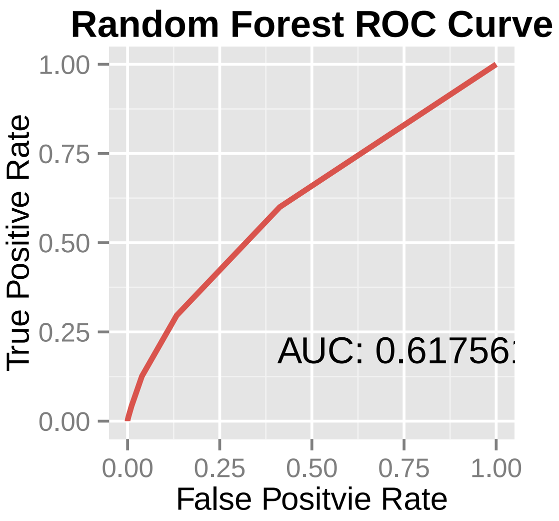


The results from the decision tree give an indication of how useful each of the variables are at predicting churn. This tree is fairly shallow, so any of the variables that made it into the tree will most likely show up in other models that give some indication of variable importance. One interesting 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.

4.3.3 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.

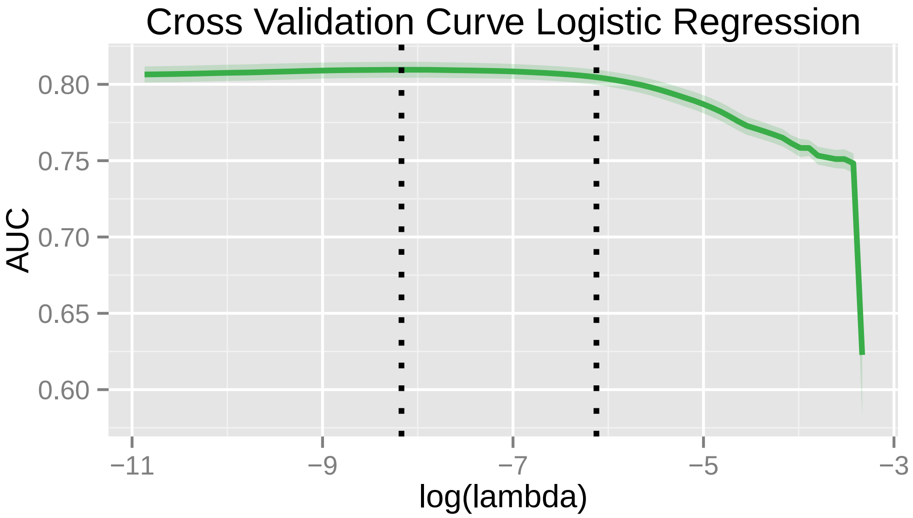
4.3.4 K-Nearest Neighbors & Naïve Bayes

The variable selection process for churn 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 or less based on the Calibration data set. The Calibration data set is a 10% random selection of observations from the original data set.

4.4 Up-Sell

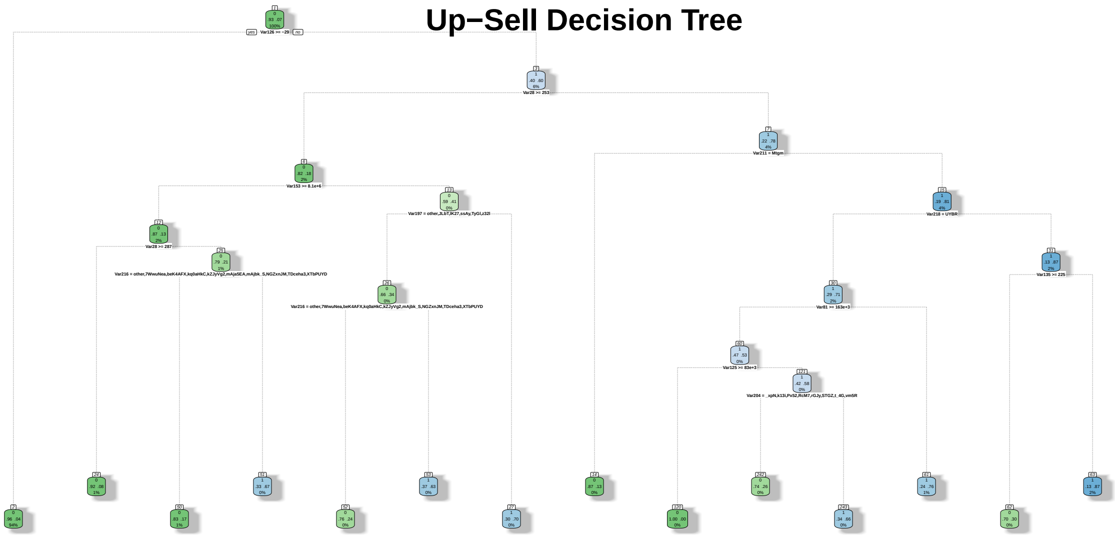
The last response variable to analyze is up-sell. As defined in the task description, up-selling can imply selling something additional, or selling something that is more profitable or otherwise preferable for the seller instead of the original sale.

4.4.1 Logistic Regression with Elastic-Net Penalty



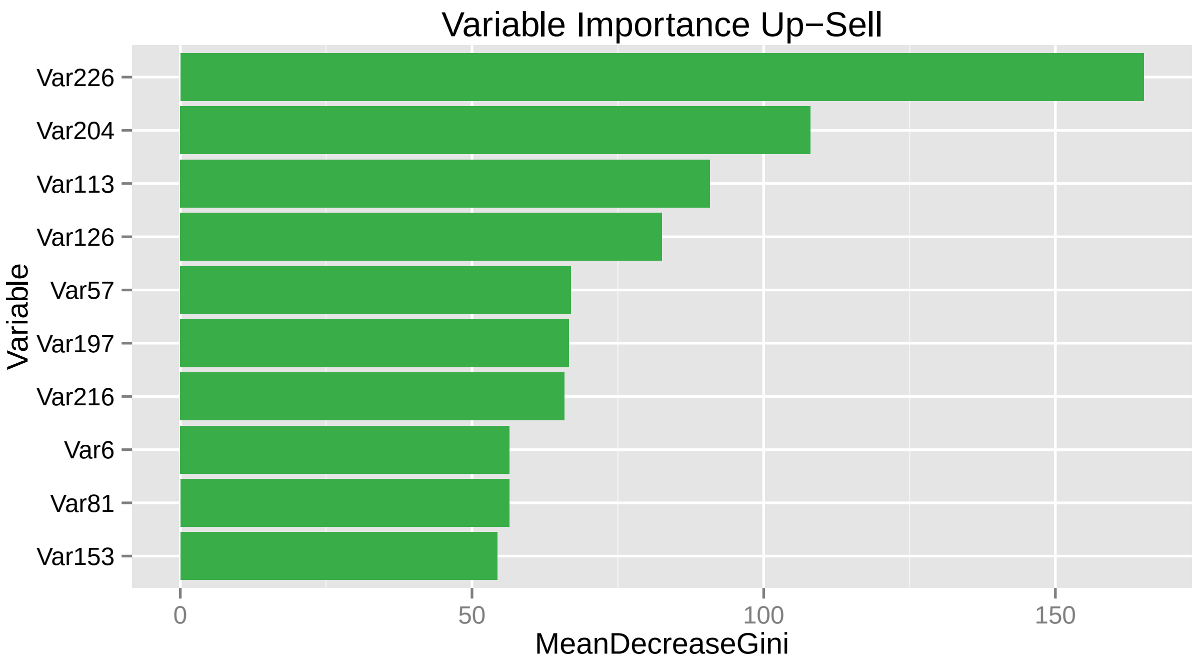
The results from the regularized logistic regression are both promising and somewhat disappointing at the same time. The plot shows that comparable performance can be achieved by removing all but about 80 variables, but unfortunately the regularization does not appear to yield much in the way of performance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

4.4.2 Decision Tree



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.

4.4.4 K-Nearest Neighbors & Naïve Bayes

The variable selection process for up-sell 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 or less based on the Calibration data set. The Calibration data set is a 10% random selection of observations from the original data set.

5 Predictive Modeling: Methods and Results

5.1 Train/Test Data

The following code was used for every model to ensure that the same testing/training data was used. We chose to use a 75/25 split for training and testing data sets.

```
# set seed to make reporducable results
set.seed(123)
# get train/test indicies
smp_size <- floor(0.75 * nrow(df))
train_ind <- sample(seq_len(nrow(df)), size = smp_size)
# split the data
train <- df[train_ind, ]
test <- df[-train_ind, ]
```

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 over fitting 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.

5.2.2.1 Random Forest 2

A second approach to random forest was attempted because online research indicated that it was a powerful algorithm for this competition. Before fitting the random forest, each positive record of appetency in the training set was copied 3 times, so that there would now be 4 copies of each positive record. This oversampling of positive appetency cases was intended to make the random forest model predict 1 for appetency more often by changing the ratio of positive and negative cases in the data. All variables were used and like many previous models missing observations were replaced with 0 and a binary variable was created to indicate missing values.

5.2.3 Logistic Regression

Logistic regression with a LASSO shrinkage was used to predict appetency. All variables were included in the model with no special transformations beyond our standard imputation of 0's, creating binary variables for missing indication and grouping the less frequent classes of qualitative variables. The value of lambda that achieved the maximum AUC on cross validated training data was used to make predictions for the test data set.

5.2.3.1 Invesitigative Varaible Selection

In addition to just allowing LASSO to select variables for logistic regression, some significant work was put into selecting variables in a more thoughtful manner with the hope that it would provide a better AUC. The process is described below.

5.2.3.2 Decision Tree Variable selection

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. Fitting a naive decision tree on the training data set produces a tree constructed using minsplit (the minimum number of observations that must exist in a node in order for a split) and minbucket (the minimum number of observations in any terminal node) is set to the values 100 and 10 respectively. We obtain that the following variables six are interesting with regards to appetency: Var126, Var204_dummy_RVjC, Var218_dummy_cJvF, Var25, Var38, and Var57.
A graphical analysis (not shown) of these 6 variables reveals the following:
1) Lower values of Var126(between -25 and +13) seem to be associated with high proportionate appetency
2) A higher count of appetency for people with no values for Var204_dummy_RVjC
3) A higher count of appetency for people with no values for Var218_dummy_cJvF
4) High counts of appetency for Var25 value below 2000
5) High counts of appetency for Var38 values below 5.0e+06
6) Relatively similar counts of appetency across all values of Var57.

5.2.3.2.1 Goodness-of-fit of decision tree variables

Using the above variables obtained from our decision tree variable selection step, we proceed to fit a logistic regression model on the entire data set using appetency as a target. We obtain that only three variables are statistically significant. An ANOVA analysis between the full model containing all 6 variables and a reduced model containing the three statistically significant variables indicates that the reduced model fits as well as the full model. The variables are Var126, Var218_dummy_cJvF and Var38. We also note a chi-square goodness of fit test for the overall model is significant at p=0.05

5.2.3.3 LASSO Variable Selection

Using the LASSO (shrinkage parameter, lambda=1) we obtain a selection of 33 variables when the shrinkage parameter, lambda, is at its minimum.

Goodness-of-fit of LASSO variables

Using the above variables obtained from our LASSO exploratory model, we proceed to fit a logistic regression model on the entire data set using appetency as a target. We find that several variables are not statistically significant, however, an ANOVA analysis between the full model containing all 33 variables identified by the LASSO, fit better than a reduced model in which statistically insignificant variables are dropped. We also note a chi-square goodness of fit test for the overall model is significant at p=0.05.

5.2.3.3.1 GOF on LASSO variables

Using the above variables obtained from our LASSO exploratory model, we proceed to fit a logistic regression model on the entire data set using appetency as a target. We find that 2 variables are inestimable and several variables are not statistically significant, however, an ANOVA analysis between the full model containing 30-2(28) variables identified by the random forest model, fit better than a reduced model in which statistically insignificant variables are dropped. We also note a chi-square goodness of fit test for the overall model is significant at p=0.05.

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 over fitting 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 up-sell to be reasonably accurate.

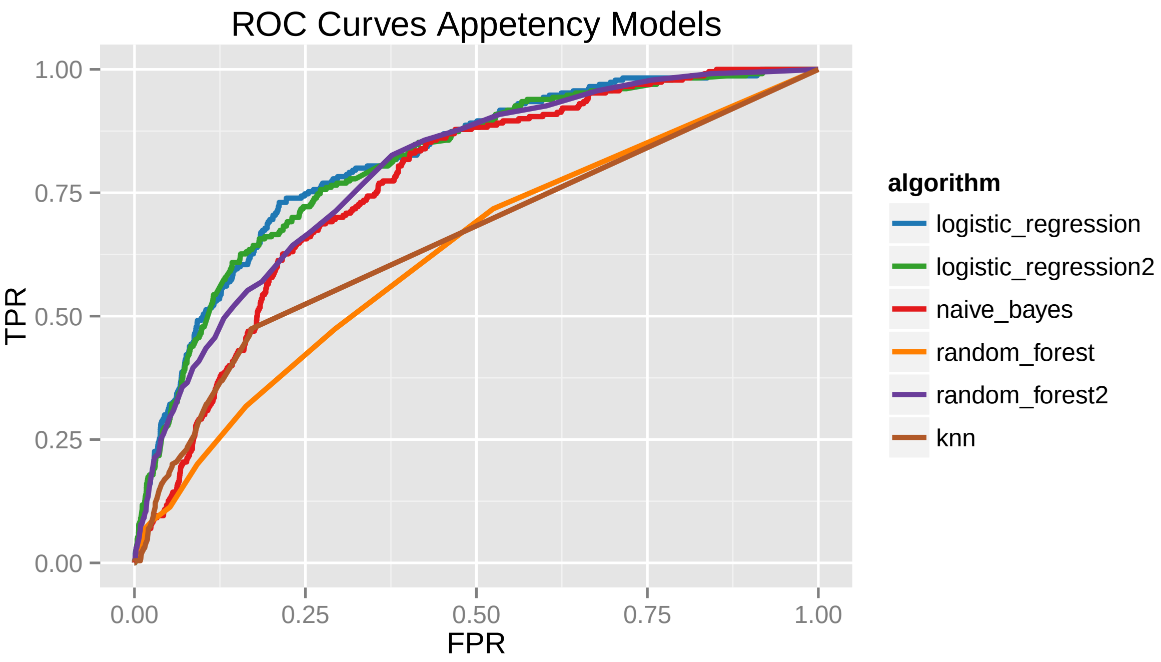
5.2.5 Support Vector Machine

The variable selection process started with the variables selected by the random forest as being the strongest and then do to performance issues, was reduced to only the 5 strongest variables.

Training the model was also taking as long as an hour for the training data set as well as even a small subset of variables, so, a separate data set of 20% of the observations was created for the SVM model.

When the results were applied to the test data set, the ROC curve shows results that don't differ significantly from a random model. The model herein relies on the radial kernel with a cost of 10. linear, polynomial, radial and sigmoid methods were all attempted without significant improvement.

5.2.6 Model Performance



	AUC
In_House	0.85
logistic_regression	0.81
logistic_regression2	0.81
naive_bayes	0.76
random_forest	0.63
random_forest2	0.79
knn	0.65

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 over fitting 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.

The model was then refined 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.

5.3.2.1 Random Forest 2

Because of the success of the second random forest attempt for detecting appetency, a similar technique was employed for detecting churn. All of the positive instances of churn were over sampled by a factor of four. This did increase the AUC for random forest, but the result was not as dramatic as it was for appetency.

5.3.3 K-Nearest Neighbors

The Nearest Neighbor K 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 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 over fitting 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 up-sell to be reasonably accurate.

5.3.4 Support Vector Machine

The variable selection process started with the variables selected by the random forest as being the strongest and then do to performance issues, was reduced to only the 5 strongest variables.

Training the model was also taking as long as an hour for the training data set as well as even a small subset of variables, so, a separate data set of 20% of the observations was created for the SVM model.

When the results were applied to the test data set, the ROC curve shows results that don't differ significantly from a random model. The model herein relies on the radial kernel with a cost of 10. linear, polynomial, radial and sigmoid methods were all attempted without significant improvement.

5.3.5 Logistic Regression

5.3.5.1 Tree Variable Selection

We started out with variables that were selected by a decision tree and proceeded to fit a logistic regression model using churn as a target. We obtain the following summary output of the model fit. We note that 12 variables have NA, which means they are inestimable. We will drop them from further consideration. We also note that a number of variables are not statistically significant. These we will also drop as well. Based on these drops, the following 9 variables are considered useful from our fitted logit regression: Var126, Var217_dummy_missing, Var211_dummy_L84s, Var73, Var126_missing, Var229_dummy_missing, Var113, Var22_missing, and Var65. Using the chi-square goodness of fit test we obtain a p-value=0.2439 and fail to reject the null hypothesis that this model is exactly correct.

5.3.5.2 LASSO Variable Selection

A listing of the variables admitted into the model at log (lambda), one standard error from the minimum are: Var7, Var73, Var113, Var126, Var22_missing, Var28_missing, Var126_missing, Var205_dummy_sJzTlal, Var206_dummy_IYzP, Var210_dummy_g5HH, Var212_dummy_NhsEn4L, Var217_dummy_other, Var218_dummy_cJvF, Var218_dummy_missing, and Var229_dummy_missing.

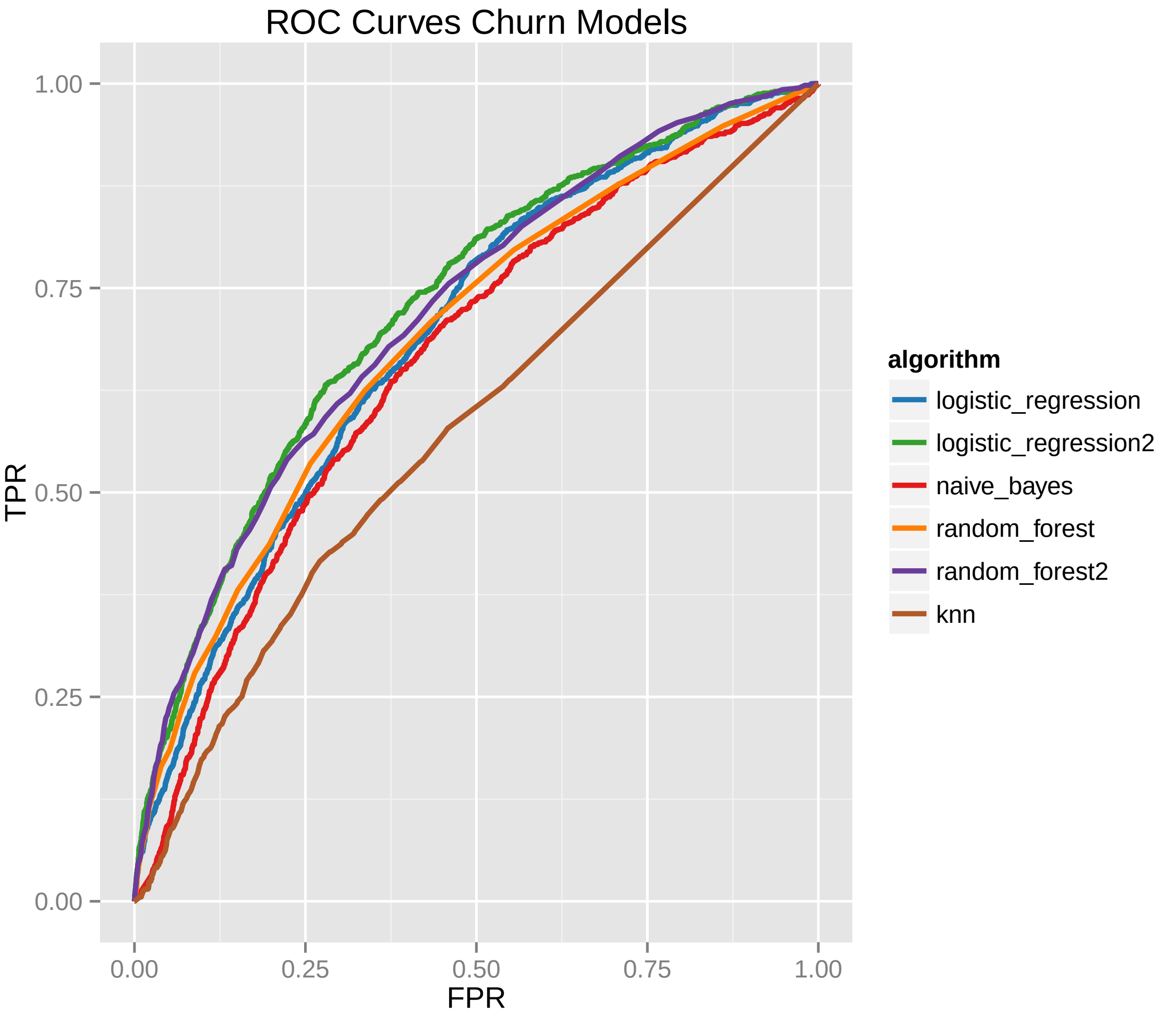
5.3.5.3 GOF on LASSO variables

Using the above variables obtained from our LASSO exploratory model, we proceed to fit a logistic regression model using churn as a target. We obtain the following summary output of the model fit. We note that 1 variable has NA, which means it is inestimable. We will drop it from further consideration. We also note that a few variables are not statistically significant. These we will also drop as well. Based on these drops, the following 10 variables are considered useful from our fitted logit regression: Var7, Var73, Var113, Var126, Var205_dummy_sJzTlal, Var210_dummy_g5HH, Var212_dummy_NhsEn4L, Var217_dummy_other, Var218_dummy_cJvF, and Var229_dummy_missing. Using the chi-square goodness of fit test we obtain a p-value=0.6240 and fail to reject the null hypothesis that this model is correct.

5.3.5.4 Simple LASSO

In addition a simple logistic regression model with LASSO shrinkage was fit. This model included all of the variables and used the standard data imputation technique that other models used. The results were on the test data set show that this set up is among the most powerful algorithms for detecting churn. In the ROC curve plot, this method is identified as logistic regression 2.

5.3.6 Model Performance



	AUC
In_House	0.74
logistic_regression	0.69
logistic_regression2	0.73
naive_bayes	0.66
random_forest	0.69
random_forest2	0.72
knn	0.58

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 up-sell.

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 over fitting 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 up-sell to be reasonably accurate.

5.4.2 Random Forest

The Random Forest classifier was used to build a classification model for up-sell 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 up-sell=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 up-sell=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 up-sell in test data is 0.8334. This is a moderate value, so the model is not a perfect fit.

5.4.3 Logistic Regression

Up-sell is the easiest of the responses to detect, but has been one of the harder problems for our team to beat or get close to the in house predictions. A logistic regression model was fitted for this problem with a LASSO shrinkage parameter. Several attempts at feature engineering were made, three interaction variables were added based on the results of the decision tree discussed in the EDA portion (variable 126 and 28, variable 28 and 153 and variable 125 and 81). Also a squared version of every single numeric variable was added to the data. This created a very large data set and the model had to be trained over a period of several hours. The results showed an improvement over some other algorithms, but failed to match the performance of the random forest algorithm.

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 up-sell.

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 over fitting 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 up-sell to be reasonably accurate.

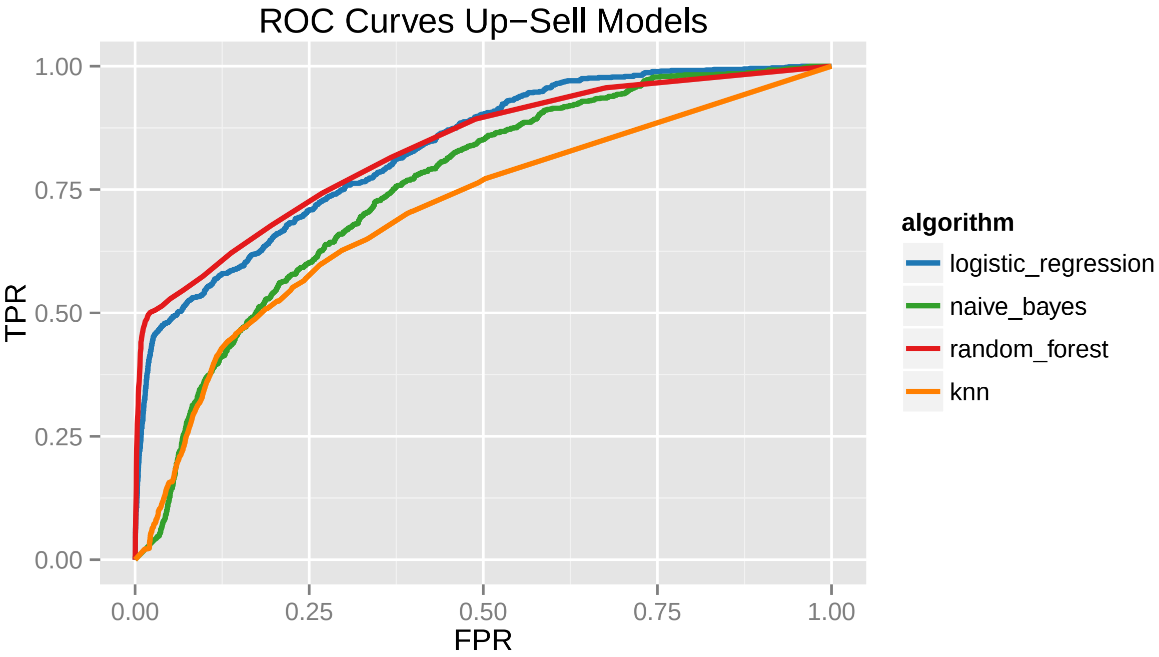
5.4.5 Support Vector Machine

The variable selection process started with the variables selected by the random forest as being the strongest and then do to performance issues, was reduced to only the 4 strongest variables.

Training the model was also taking as long as an hour for the training data set as well as even a small subset of variables, so, a separate data set of 20% of the observations was created for the SVM model.

When the results were applied to the test data set, the ROC curve shows results that are slightly better than a random model. The model herein relies on the radial kernel with a cost of 10. linear, polynomial, radial and sigmoid methods were all attempted without significant improvement. An automate method was explored for identifying the appropriate level of C (cost) and gamma however due to the size of the data set was not successful. Cost was manually adjusted using a range from 1 indicating the narrowest margin on the hyper-plane to 10000 indicating a very wide hyper-plane with many miss-classifications.

5.4.6 Model Performance



	AUC
In_House	0.90
logistic_regression	0.83
naive_bayes	0.75
random_forest	0.83
knn	0.70

6 Comparison of Results

7 Conclusions

8 Appendix

We would have liked to include the R code to produce the results described in this paper, but amount of code is prohibitively long. For the interested reader, all of the code used to create our results can be found at: https://github.com/jayswinney/454-kdd2009