

Ensemble Approach to Predict Churn, Appetency and Up-Sell

Srividhya Kannan

srividhya.kannan@latentview.com

LatentView Analytics

JVL Plaza, Ground Floor

626, Anna Salai, Teynampet

Chennai 600 018, India

Ph: +91-44-4509 4040 / 4039

Priya Balakrishnan

priya.balakrishnan@latentview.com

LatentView Analytics

JVL Plaza, Ground Floor

626, Anna Salai, Teynampet

Chennai 600 018, India

Ph: +91-44-4509 4040 / 4039

Editor: Vivek Desikan, Pramad Jandhyala, Krishnan Venkata, Padmasini Rengabashyam

Abstract

The abstract describes various approaches followed by LatentView Analytics to develop fast scoring solution on a large telecommunication database. ***Over the course of the challenge, we developed around 160 models using one or more techniques.*** Our over-all approach in developing these models and our final solution can be summarized as ***“Ensemble of Models developed using suite of Logistic regression models, Gradient Boosting, Adaptive Logistic Regression, Decision Tree and Naive Bayes algorithm on various random samples and using innovative techniques to combine the predictions from various models to develop the final score.”*** The detailed summary follows.

Challenge

The challenge was to build three solutions that would predict the propensity of the customer to buy another product (Appetency), buy a product of higher value (Up-sell) and switch provider (Churn) in the telecommunications context. Key feature of the challenge was to develop a **fast scoring solution on a**



large database within five days from the release of the data. The challenge is to beat the in-house system developed by Orange Labs. It was an opportunity to prove our abilities to deal with a very large database, including heterogeneous noisy data (numerical and categorical variables), and unbalanced class distributions. Time efficiency to deliver a quick solution was also evaluated.

Data

Data was provided by Orange Labs, one of the largest telecommunication operators in the world. The organizers provided a training sample to develop the models and a test sample was provided for evaluation of the models. The target variables were provided only for the training sample. Key attributes of the tables provided are shown below:

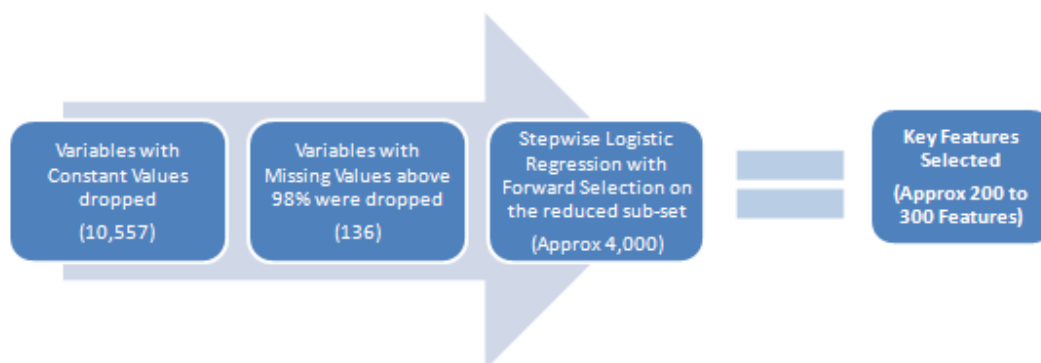
Attributes	Training Sample	Test Sample
Total Records	50,000	50,000
Numeric Feature Variables	14,770	14,770
Character Feature Variables	230	230
Total Feature Variables	15,000	15,000

Target Variable	Training Sample
Churn Rate	7.34%
Appetency Rate	1.78%
Up-sell Rate	7.36%

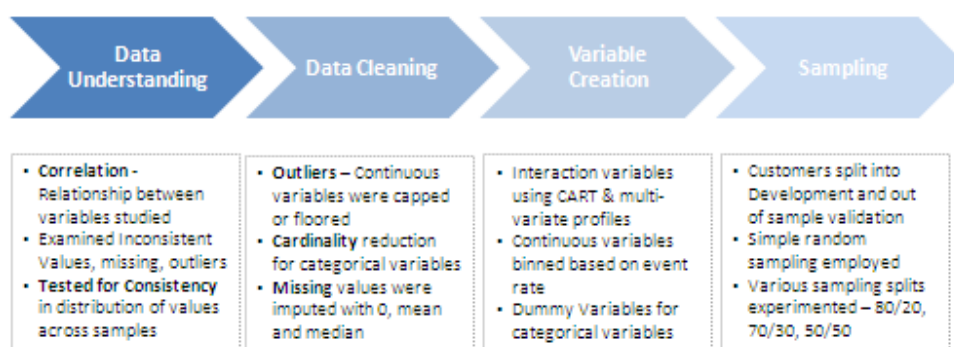
Data Understanding and Data Preparation

As the number of features were very large, it was critical to perform feature selection to reduce the data to include usable and important features. Variable names were encrypted for security reasons and hence interpretability of the variables was not possible.

We followed various approaches listed below to subset the key features from a list of 15,000 variables for model development process



For each of the three models, approximately 200 to 300 features were identified using the approach detailed above. Once the key features are identified, data was prepared for developing models. The data preparation process is illustrated in the graphic below:



Correlation: between the variables were studied and among the highly correlated variables, one of the variables was retained.

Consistency between Samples: We compared the frequency distribution of categorical variables and uni-variate of numeric variables to make sure the distributions are not different. This was done to ensure stability of variables over time. We found the distribution to be similar for all the variables.

Outliers: We examined various outlier treatments like capping at the 99th percentile for every continuous variable and flooring at the 1st percentile and values outside $\pm 3\sigma$. However, we finally chose to cap and floor the extreme values at the above mentioned percentiles as the later was a more stringent criterion imposed for outlier treatment.

Cardinality Reduction: Categorical variables with more than five categories were grouped. After grouping, the categories with less than 2% observations were further grouped.



Missing Treatment: We explored three different missing value treatments for the numerical variables short-listed as key features. We imputed the missing values with 0, mean and median and provided these as inputs to our models. We let the model choose the best imputation for the variables.

Interaction Variables – There are several ways in which we created interaction variables. A few are listed below

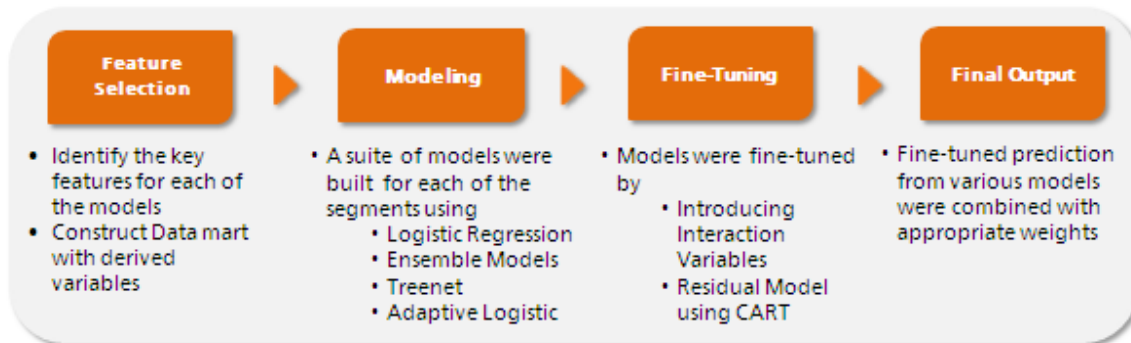
- a. Indicators for two dimensional interaction variables were created from CART. The interaction variables were identified based on the variables entering the top split
- b. Variables were profile with the outcome variable and plotted to observe any significant difference among the levels of one variable for change in the levels of other variable. Interaction variables were created for pair of variables wherever significant difference was observed.

Experiments

Advanced statistical approaches like like logistic regression, CART models, Adaptive logistic regression, SVM, Gradient boosting algorithm, Naïve Bayes algorithm were evaluated as part of solution development process. Based on the different experiments performed, we deduced that –

- a. No single approach was performing well on the entire population. Each technique has its own advantages and limitations. This could be due to various reasons like *availability of enough data points* to build a stable model using techniques like TreeNet, *non-linear effects* present in the data where a TreeNet model may be better than Logistic regression.
- b. Estimates from models based on one sample was not stable. This could be due to fact that the sample was not representative of all the characteristics in the population. In order to get a true representation of the population, we chose to build on various samples
- c. Models based on small samples but **large number** of samples were boosting the performance as various samples may possess different characteristics which could be captured in the model
- d. Linear model performance could be improved by predicting the residuals using non-linear modeling approach. The residuals could be due to the non-linearities that are not captured in a model. Thus, a non-linear model to capture the residual effect and the predicted score adjusted for the residual is expected to improve the performance.

On basis of the above interpretations, we finalized the final approach to develop the solution and they are broadly illustrated in the four steps below:



LatentView's Detailed Solution Methodology

Step by step approach that we adopted is enlisted below

Step I - Logistic Regression Model

- **Single Logistic Model**
 - For each of the target variables, stepwise logistic model was built on the 50,000 records using the key features identified and the derived variables. We used the log-link function to develop this model
- **Ensemble Logistic Model**
 - Generate 50 random samples of 30,000 records, say S_1, S_2, \dots, S_{50}
 - For each of the samples, we developed stepwise logistic model with the key features identified in the single logistic model using log-link function. Let the probabilities be P_1, P_2, \dots, P_{50}
 - The probabilities were combined to obtain the ensemble logistic score using methodologies like
 - $Average(P_1, P_2, \dots, P_{50})$
 - $Median(P_1, P_2, \dots, P_{50})$
 - $Min(P_1, P_2, \dots, P_{50})$
 - $Max(P_1, P_2, \dots, P_{50})$
 - Average of the estimates from the 50 models were derived and the scores were computed based on the new estimates



- Median of the estimates from the 50 models were derived and the scores were computed based on the new estimates
- Logistic model was built with the actual outcome as the DV and P_1, P_2, \dots, P_{50} as independent variables in order to eliminate the impact of non-representative samples

Thus, we have 8 scores based on the logistic model (1 based on single model and 7 based on ensemble of models). Let us denote these scores as $L_1, L_2, L_3, \dots, L_8$

Step II – Decision Trees

As logistic regression has the limitations of capturing non-linearities, we used decision tree algorithms to capture non-linearities

- **TreeNet Model**

- Develop TreeNet models with the key features selected to predict the outcome
- We built around 600 trees on 80% of the sample and validated TreeNet scores on the remaining 20% sample
- We used TreeNet software to develop these models
- Let us denote the TreeNet score as T_1

- **CART Model**

- Develop CART models with the key features selected to predict the outcome
- We developed CART models on 100% training sample
- We used CART software and R to develop these models
- Let us denote the CART score as C_1

Step III – Adaptive Logistic Regression and Naive Bayes Algorithm

We used Gains # software to develop Adaptive Logistic Regression score (ATS Score) and Naïve Bayes Score. ATS uses non-linear functional form to fit the relationship between the outcome variable and the independent variables. We used the key features identified as input for both the models. Gains # has ways to do the feature selection and then build the model on the most important variables.

Let us denote the ATS Score as A_1 and Naive Bayes Score as NB_1 .

Step IV – Residual Model

This is one of the concepts internally developed at LatentView to account for effects not captured in the model. We developed a residual model for Logistic Model scores (L_1 to L_8), ATS Score (A_1) and Naive Bayes score (NB_1). We used TreeNet to develop the residual model. Detailed steps are described below:

- The outcome variable for the residual model would be (Actual Outcome – Predicted Score). The outcome variable would range between -1 to +1
- The independent variables would be the key features. The hypothesis is that the features that got selected in the individual Logistic or Adaptive Logistic or Naive Bayes models would have least role to play in the residual models compared to features that did not get selected in the individual models.
- Develop TreeNet Models to predict the residual
- Residual Improved Score = Predicted Score + Predicted Residual
- Notations
 - Residual Improved Logistic Scores are denoted as RL_1, RL_2, \dots, RL_8
 - Residual Improved Adaptive Logistic Score as RAL_1
 - Residual Improved Naive Bayes Score as RNB_1

Step V – Joint Score

The last step was to combine the predicted scores from various techniques to obtain the final prediction for each of the models. Various weighting approaches were followed as described below

- Simple average of scores from various techniques. For eg, $L_1, T_1, AL_1, NB_1, C_1$
- Weighted average where weights are user defined based on experience in application of these techniques
- Minimum of scores
- Maximum of scores
- Weighted average of scores where weights were determined based on Logistic model. The dependent variable was the outcome variable and the predicted score from various techniques were used as independent variables
- Segmented Logistic Regression model to determine the weights for each score where two scores are to be combined



- We ran CART models with $\text{Abs}(\text{Score}_1 - \text{Score}_2)$ as the dependent variable and other key features as independent variables
- Individual logistic regression models were built in the segments identified in the top split. The concept is that single technique need not apply across all customers.

Weighted average described above is computed for various combinations of scores obtained from different techniques. For eg, weighted average of Logistic, TreeNet & CART Score, weighted average of Logistic and TreeNet Score, etc where Logistic score could be $L_1, L_2, \dots, L_8, RL_1, RL_2, \dots, RL_8$.

Applications Used

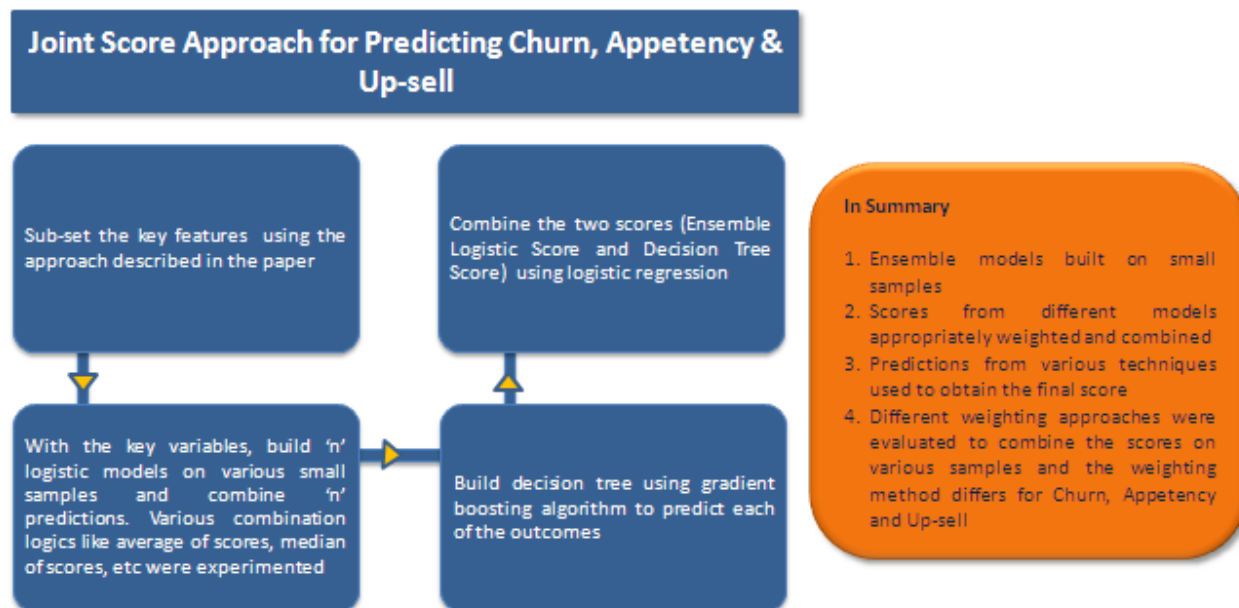
We explored various freeware and third party software for developing the models. The final solution used the following applications -

1. Logistic Models - SAS
2. TreeNet models - Salford Systems
3. CART models - Salford Systems and R Software
4. Adaptive Logistic Regression and Naïve Bayes algorithm were built using a freeware (Gains #)

Solution was developed using automated programs and independent modules were developed in parallel.

LatentView's Final Solution Methodology

The individual models, and all permutations and combinations of the scores using various techniques gave rise to 160 models. We rank ordered the models based on the training performance (AUC) and the test performance and chose the best model. The composition of our final solution is illustrated below:



Results

The final solution that won the fourth position in the fast track competition was

- **Up-sell** – Score based on TreeNet model
- **Appetency** - Weighted average of Ensemble score based on Average of 50 predicted scores and TreeNet model where the weights for the Logistic score and TreeNet score were determined by a simple logistic regression
- **Churn** - Weighted average of Ensemble score based on Average of estimates of the 50 models and TreeNet model where the weights for the Logistic score and TreeNet score were determined by a simple logistic regression

Acknowledgement

We would like to acknowledge the support provided by LatentView Analytics in making this participation possible. We would like to thank Venkat Viswanathan, CEO of LatentView Analytics, Pramad Jandhyala, Director of LatentView Analytics for their encouragement which helped us secure a creditable fourth position in the prestigious KDD Cup, 2009. We thank the editors of this paper and colleagues for their valuable suggestions. We also thank the organizers of the KDD cup for a challenging contest and their extended support in responding to our queries.