Churn Modeling Project

By: Biljana Jovanova and Rajesh Chodavarapu

Contents

[Contents 1](#_Toc520967715)

[Introduction to Problem Statement: 3](#_Toc520967716)

[How can predictive modeling help companies in their retention efforts? 3](#_Toc520967717)

[Our learning goals from this project: 3](#_Toc520967718)

[Original Dataset: 3](#_Toc520967719)

[The Accuracy Paradox- Choosing the Performance Metric. 4](#_Toc520967720)

[Modeling Methodology: 5](#_Toc520967721)

[Data Splitting Technique: 7](#_Toc520967722)

[Resampling technique: 7](#_Toc520967723)

[Exploratory Analysis: 7](#_Toc520967724)

[Training Datasets: 7](#_Toc520967725)

[Predictive Modeling 7](#_Toc520967726)

[Model 1: Random Forrest 8](#_Toc520967727)

[Predictive Power 8](#_Toc520967728)

[Interpretability 9](#_Toc520967729)

[Model 2: Decision Tree 12](#_Toc520967730)

[Predictive Power 12](#_Toc520967731)

[Interpretability 14](#_Toc520967732)

[Comparing the Random Forrest vs Decision Tree 15](#_Toc520967733)

[Model 4: Quadratic Discriminant Analysis (QDA) 15](#_Toc520967734)

[Predictive Power 15](#_Toc520967735)

[Interpretability 16](#_Toc520967736)

[Model 5: Liner Discriminant Analysis (LDA) 16](#_Toc520967737)

[Predictive Power 16](#_Toc520967738)

[Model 5: Logistic Regression 18](#_Toc520967739)

[Predictive Power 18](#_Toc520967740)

[Interpretability 19](#_Toc520967741)

[Model 6: Logistic Regression based on Youden’s J Index 19](#_Toc520967742)

[Predictive Power 19](#_Toc520967743)

[Interpretability 20](#_Toc520967744)

[SUMMARY ON CLASSIFICATION MODELS 20](#_Toc520967745)

[Ensemble Models 20](#_Toc520967746)

[Types of ensembling 21](#_Toc520967747)

[Results of Ensembling 22](#_Toc520967748)

[Advantages and Disadvantages of ensembling 22](#_Toc520967749)

# Introduction to Problem Statement:

The Telecommunications business model relies on a combination between subscription and usage revenue models. In a subscription model, customers sign a contract and pay a set fee for on recurring basis until a contract expires or customer defects voluntarily, in effect ending the continual stream of recurring revenue in perpetuity. For a utility model, the customer pays a fee based on the amount of usage within a monthly period, which can be different from one month to the next. For the telecommunication provider there are typically Capital Expenditure (CapEx), or fixed costs, related to physical devices which the customer uses. Traditionally, between 10-20% of the providers’ revenue go to building, upgrading, and maintaining their operational and network environment[[1]](#footnote-1). Due to depreciation schedules of fixed assets, **keeping existing customers on existing gear makes the customer more profitable in the long run.**

Furthermore**, acquiring new customers can be very expensive** due to direct and indirect costs of acquisition of the same, including but not limited to marketing and sales. The digital market is increasingly more competitive[[2]](#footnote-2) which in turn makes the buying power of customers high, driving management to place **a high priority on retaining existing customers, or reducing defection rates**.

Retention models are also called churn models, and they use historic customer behavior as measured in different variables with the goal to assign a risk profile or probability of churn. Collecting relevant data related to the customer usage and interaction behavior is paramount.

## How can predictive modeling help companies in their retention efforts?

In churn models it is important not only to predict whether a customer will churn or not, but also what *drives* churn. Understanding the key drivers can help management focus on creating operational strategies change the defection pattern and mitigate churn.

The goal should always be to build a model that is both accurate in predicting the churn event, but also interpretable.

## Our learning goals from this project:

Our primary goal is to apply our techniques acquired in the program, and build classification models on a open source dataset to answer a real-world business problem. Secondarily, our interest was to learn and apply new techniques for dealing with imbalanced classes in the response variable and study its impact on the accuracy of the predicting the event and the overall accuracy of the model.

## Original Dataset:

Source: Telcom Churn dataset from Kaggle website.

Dimensions: 20 predictor variables and one response (Churn), 5000 observations

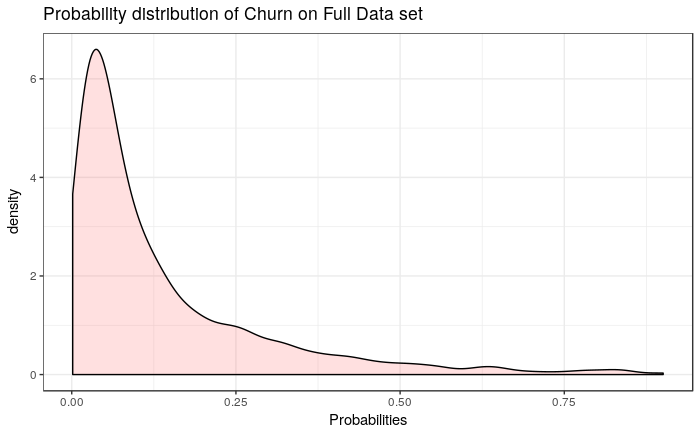


Response frequency is unbalanced: Churn | yes = 14% and Churn | no = 86%

## The Accuracy Paradox- Choosing the Performance Metric.

Predictive power relates to the power of the model to accurately predict for the problem of interest we want to model, in our case probability of positive churn event. The accuracy paradox for predictive modeling states that predictive models with a given level of sensitivity or specificity may have greater predictive power than models with higher overall accuracy.[[3]](#footnote-3)

The accuracy paradox is especially true when the response variable classes have a severe imbalance. One consequence of this is that default models usually produce a classification boundary at the 0.5 probability level. Thus, the performance is generally very biased against the class with the smallest frequencies. For example, if the data have a majority of observations belonging to the first or the nonevent class and a small percent fall in the second or event class, most predictive models will maximize accuracy by predicting everything to be the nonevent class, therefore the model will achieve high specificity but very poor sensitivity.

For example, the chart above shows the predicted probabilities for churn | yes. We can see that most observations are around 10-15% probability, thus classifying based on 0.5 decision boundary seems quite arbitrary and will identify a small amount of customers as churn.

For that reason our goal is to maximize accuracy in the churn event “yes” class, by choosing the model or method which maximizes **SENSITIVITY.**

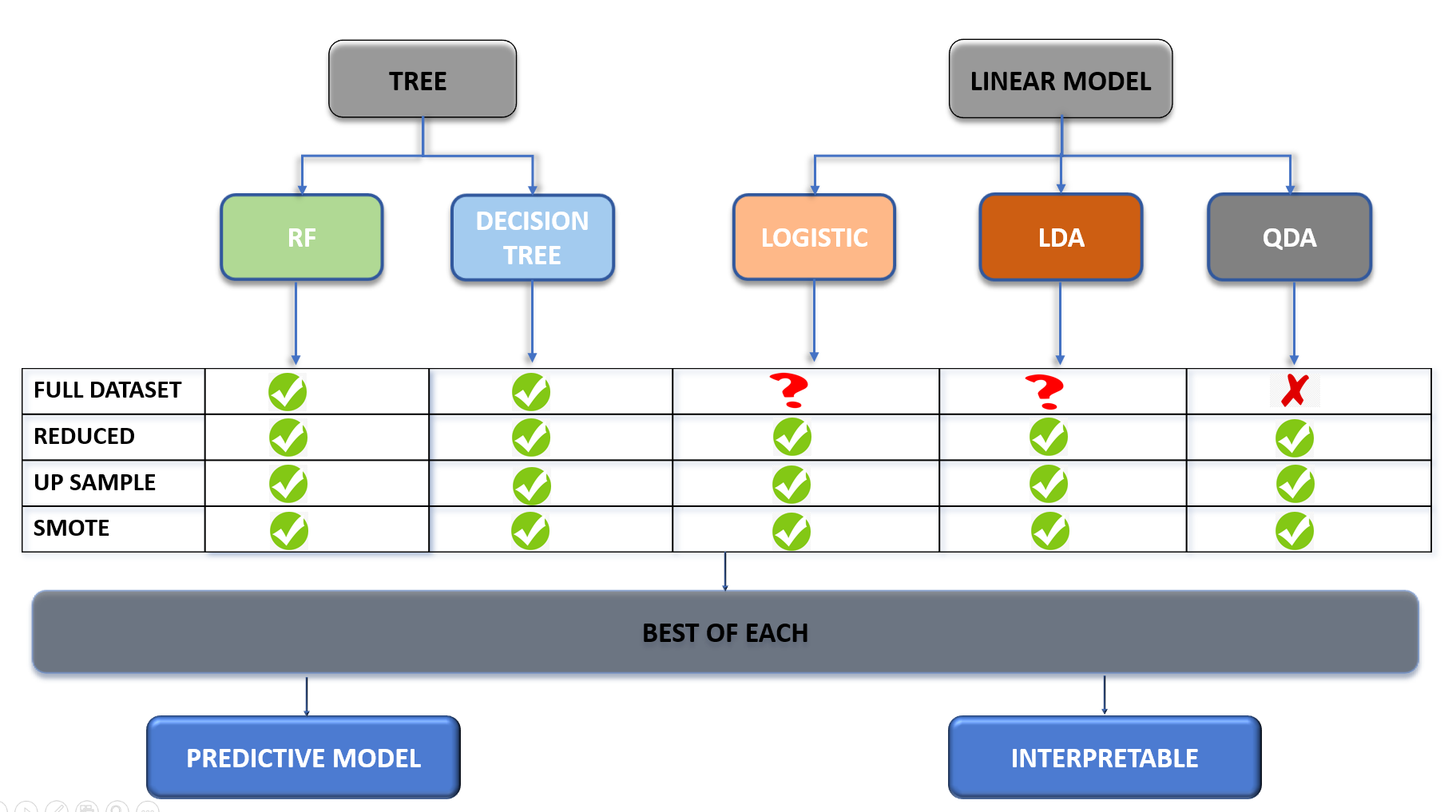
We will also compare every model based on the **Kappa Statistic** which considers the class distributions by taking into account the accuracy that would be generated simply by chance. It takes values ranging between −1 and 1, with a value of 0 means there is no agreement between the observed and predicted classes, while a value of 1 indicates perfect agreement of the model prediction and the observed classes. Good concordance is specific to the use case, however a decent Kappa value ranges between .3 to .5.

To study the effect on performance metrics from out of the box, default models with a 0.5 probability threshold, we will compare them to custom models based on different probability threshold as given by **Youden’s J metric**. This metric measures the proportions of correctly predicted samples for both the event and nonevent groups, and given different proportions we can find the single threshold which maximizes the sensitivity metric.

# Modeling Methodology:

We will use the methodology for choosing the best model based on Kuhn, the author of the Applied Predictive Modeling book.

1. Start with a model which is least interpretable and most flexible, such as Random Forrest. Across many problem domains, these models have a high likelihood of producing the empirically optimum results (i.e., most accurate).
2. Investigate simpler models such as Decision Tree based on CART, LDA, QDA, and Logistic Regression
3. Choose the simplest, most interpretable model which reasonably approximates the accuracy of the more complex models.



To study the effect of the class imbalance we will:

1. Build all models with out of the box threshold of 0.5
2. Use sampling techniques such as up-sampling and SMOTE
3. For Logistic Regression, build models based on the best Youden’s J which maximizes the sensitivity metric

For each model we will calculate:

* Overall Accuracy as measured by the Area Under the Curve (AUC)
* Sensitivity
* Specificity
* Kappa statistic
* Variable Importance metrics

**However, we will choose the best models based on Sensitivity and Kappa.**

From the best models find build an **ensemble model,** to see if we can increase the prediction power for sensitivity metric.

# Data Splitting Technique:

Prior to the exploratory analysis we first performed data splitting, to shield our test data exclusively for model validation purposes. Our training sets included 75% of the data and 25% were reserved for testing.

## Resampling technique:

We used 10 k-fold cross validation with 5 repeats to fit models and find the best tuning parameter for the models. This technique is appropriate with our overall sample size, it has a good control over bias and variance, and was computationally efficient.

# Exploratory Analysis:

The exploratory analysis on the training set showed:

1. Correlations, distributions, association between predictors and response
   1. High Correlations (Total Day Minutes vs Total Day Charge; Total Eve Minutes bs Total Eve Charge; Total Night Minutes vs Total Night Charge; Total Intl Minutes vs Total Intl Charge; Total Charge vs Total day minutes; Total Charge vs Total Day Charge;
2. Unbalanced predictors (International Plan (no- 91%, yes – 9%); Voicemail Plan (no- 74%, yes- 26%)
3. Unbalanced response (churn (no – 86%, yes- 14%)
4. Zero Variance columns -number vmail messages
5. Feature engineering: pricing components, are different customers paying more? Calculate the call rates by dividing total charge by total minutes (looks like there is not pricing difference between customers for day rate, evening rate, night rate, and international rate) the rates are different overall international being the most expensive, followed by day and evening rate, and night call rate being the cheapest one.
   1. No need for transformations to address distributional concerns
6. Uninformative variables (phone number, area code, state (has too many levels, glm models cap performance at 12 factor levels))

# Training Datasets:

1. Full Dataset: original dataset
2. Reduced Dataset: original dataset without zero variance and highly correlated predictors
3. Upsampled Dataset: reduced dataset Upsampled
4. SMOTE Dataset: reduced dataset Upsampled

# Predictive Modeling

Two separate class of models were fit on the four training datasets:

* Tree Based:
  + Random Forrest (tuning parameter mtry )
  + Decision Tree (tuning parameter – number of final terminal nodes(pruning))
* Linear Models:
  + Quadratic Discriminant Analysis (QDA)
  + Linear Discriminant Analysis (LDA)
  + Logistic Regression
  + Logistic Regression with manually adjusted probability threshold as per Youden’s J index

**The 24 different models were compared based on Sensitivity and Kappa, because we want a model which gives us the highest predictive power to identify the churners (Churn | yes class).** Furthermore, we will also examine the interpretability of each model and note which variables are the drivers behind churn.

## Model 1: Random Forrest

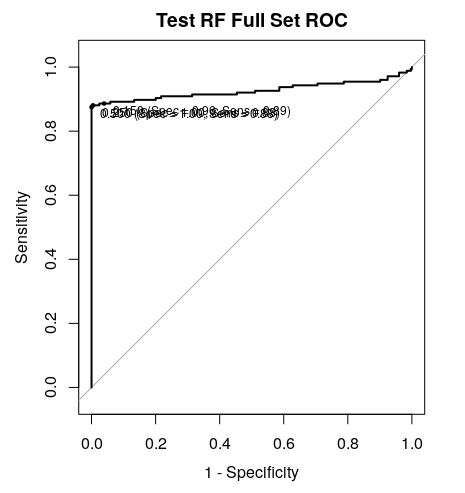
### Predictive Power

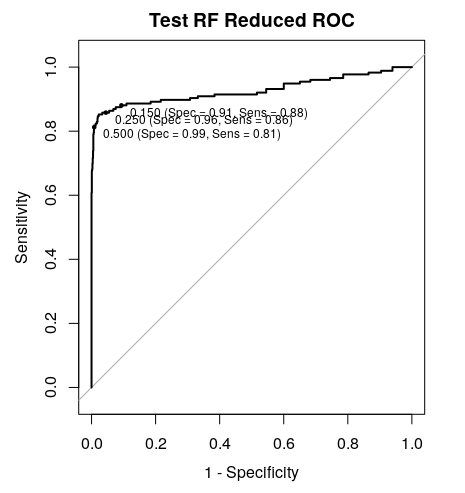
Three mtry tuning parameters were fit on the Random Forrest model with values of 4, 5, and 6. Mtry simply tells the random forest algorithm that only 4, 5, or 6 predictors should be considered for each split of the trees. Using a small value of m in building a random forest will typically be helpful when we have many correlated predictors. Using 10 k-fold resampling cross validation technique, the final Random Forrest models were fit using 5 predictors at each split.

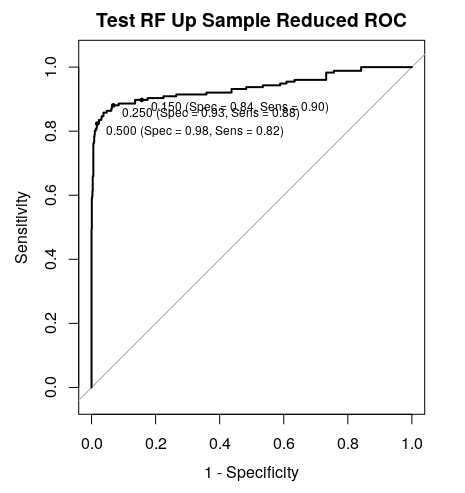
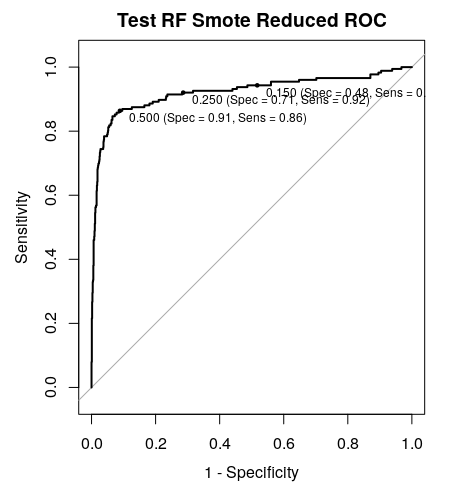
Random Forest uses probability threshold of 0.5 for classification purposes.

The best test results were achieved on the full training dataset with all the predictors. Sensitivity suffered slightly when we fit the same model on the reduced dataset, and it seems like Upsampling and SMOTE did not provide improvements. The test Kappa on the full training set was very high too.

Based on the results, we conclude that the class imbalance was not an issue for Random Forrest.







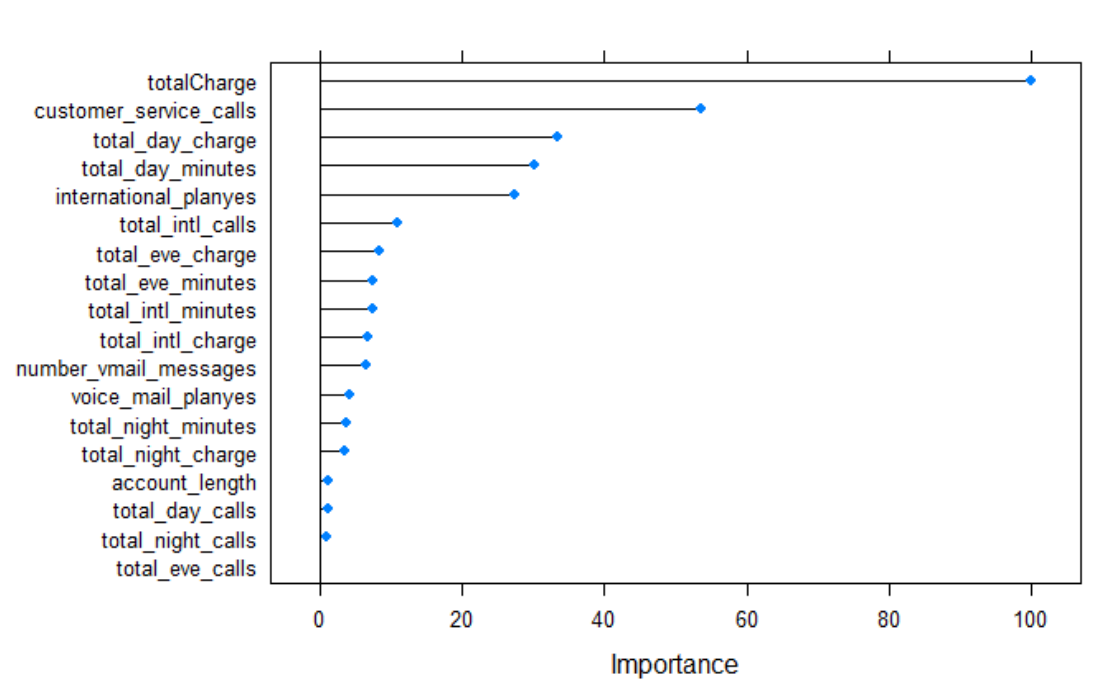


### Interpretability

Most literature on random forests and interpretable models would lead you to believe this is nigh impossible, since random forests are typically treated as a black box. Indeed, a forest consists of a large number of deep trees, where each tree is trained on bagged data using random selection of features, so gaining a full understanding of the decision process by examining each individual tree is infeasible. Furthermore, even if we are to examine just a single tree, it is only feasible in the case where it has a small depth and low number of features. A tree of depth 10 can already have thousands of nodes, meaning that using it as an explanatory model is almost impossible.

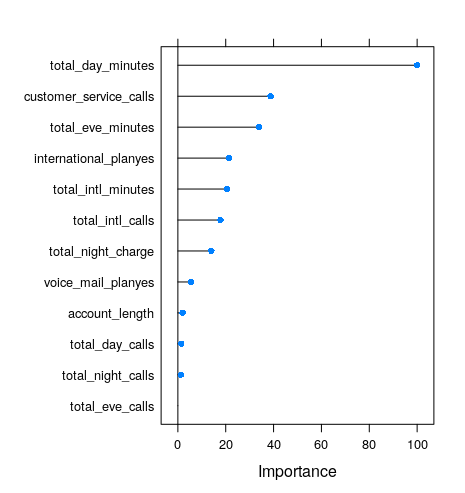
One way of getting an insight into a random forest is to compute feature importance. In caret Variable importance[[4]](#footnote-4) is calculated as “For each tree, the prediction accuracy on the out-of-bag portion of the data is recorded. Then the same is done after permuting each predictor variable. The difference between the two accuracies are then averaged over all trees and normalized by the standard error. “

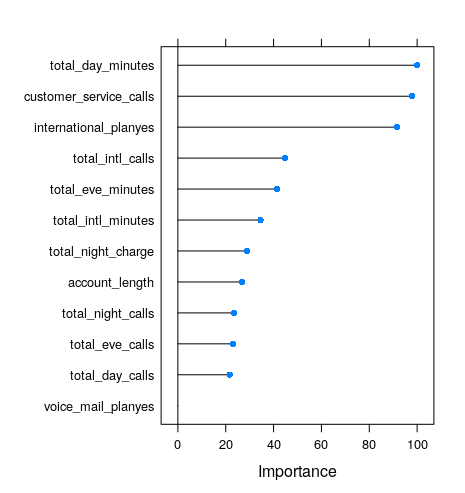
**RF Full Data Set Variable Importance Plot**



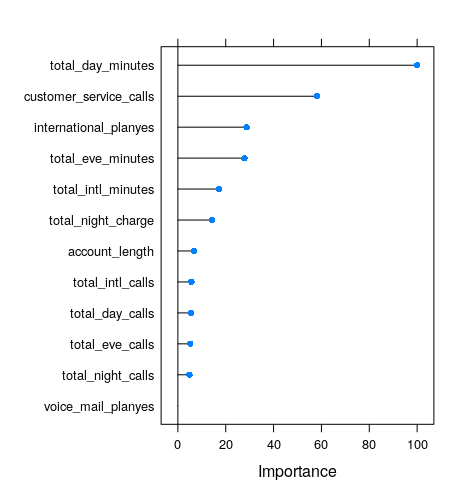
The top three most important variables are Total Charge, Customer Service Calls, and Total Day Charge.

RF Reduced Data Set Variable Importance Plot



RF Smote Reduced Variable Importance plot

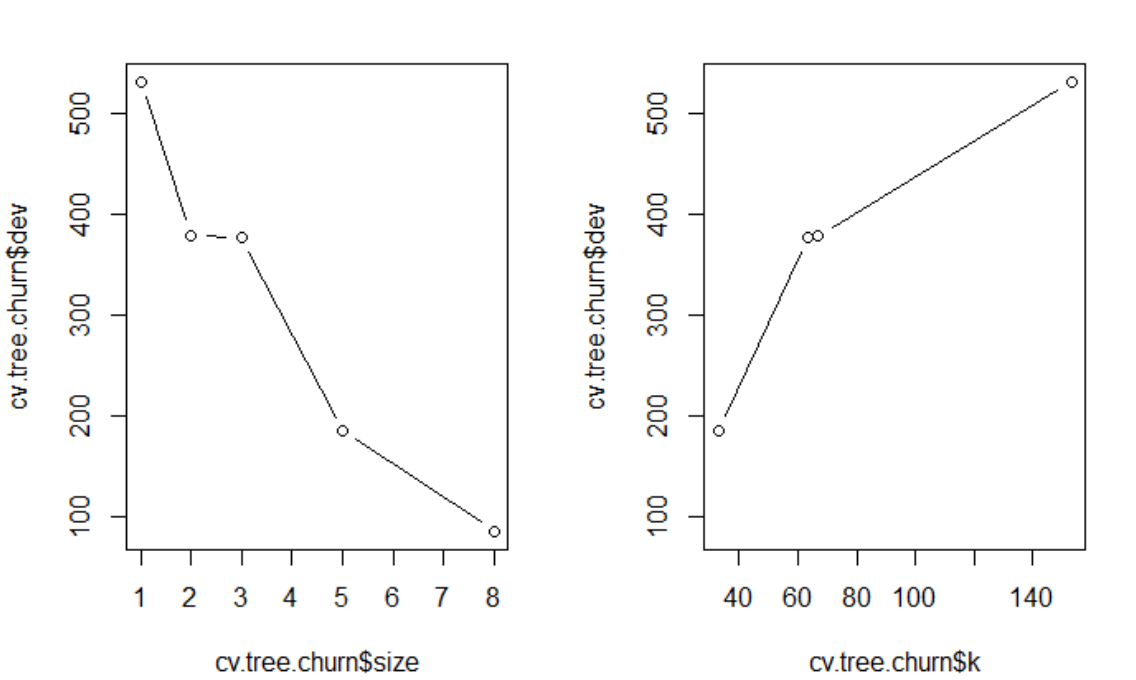
RF Up Sample Reduced Variable Importance plot



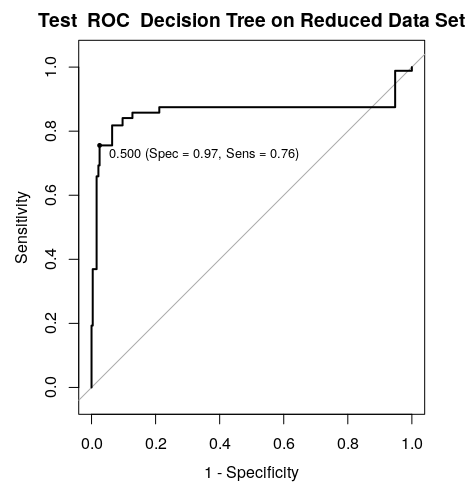
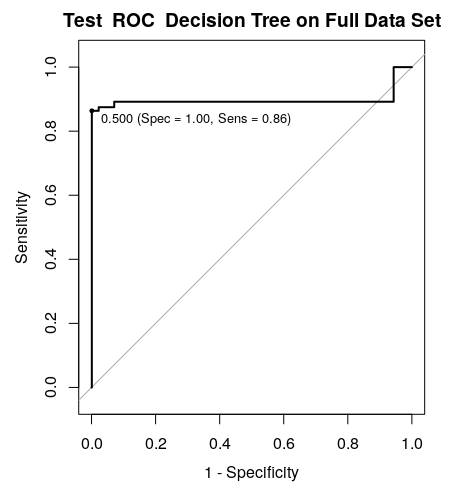
## Model 2: Decision Tree

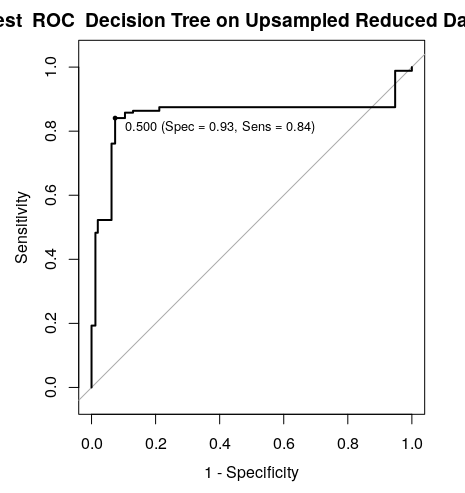
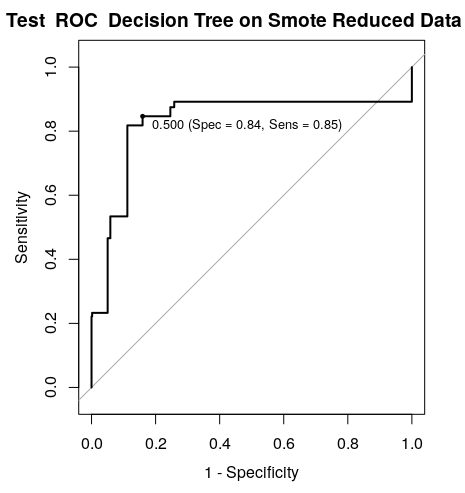
### Predictive Power

A large Decision Tree was fit on the data for all training sets. Subsequently, it was pruned based to find the best number of terminal nodes.



The lowest deviance was achieved when the tree size had 8 terminal nodes. We then fit 8 terminal node decision tree for all training sets.

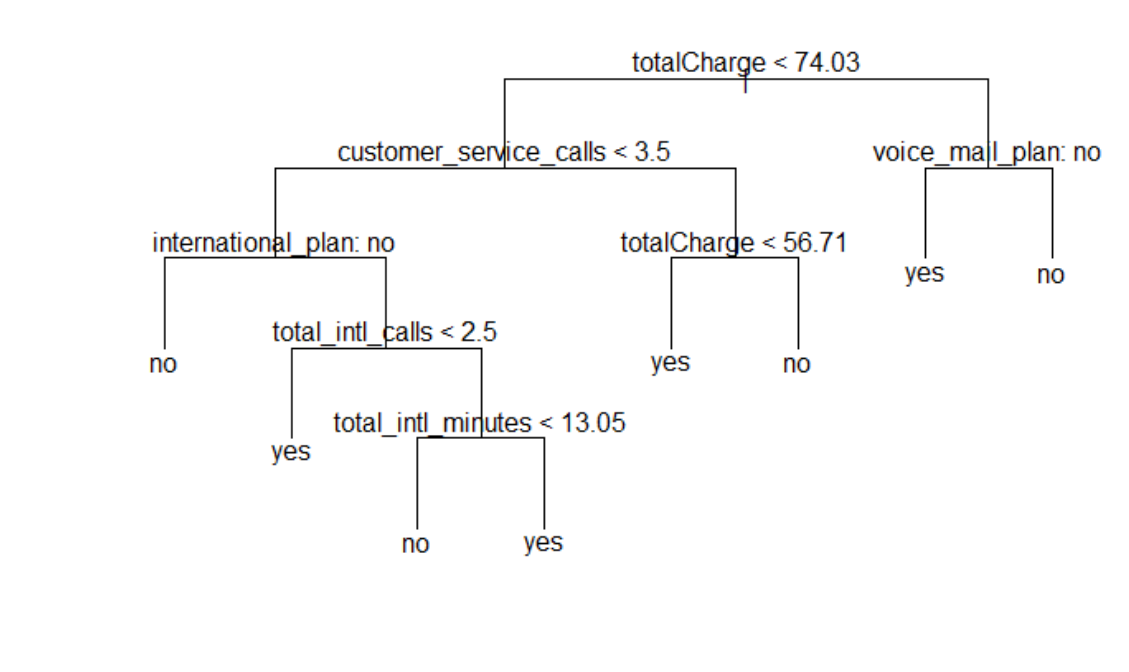






The best performance was achieved on the full training set.

### Interpretability



The decision tree plot shows us that for the best model the most important variable is Total Charge.

Decision lines which result in Churn | yes

* If the customer is paying more than $74 and does not have a voice plan
* If customer is paying less than $74, has had less than 3 service calls, has an international plan but has made less than 2 international calls
* If customer is paying less than $74, has had less than 3 service calls, has an international plan but has made more than 2 international calls, and total number has more than 13 international minutes.

This is very useful. We see a common pattern with international callers. It seems like customers who make a lot of international calls have high charges as well. Maybe we can offer a certain discount to retain them. Also, it seems like customers who have made less than two international calls and have no international plan are predicted to churn. We can offer them a promotional international plan or if they made the call by mistake maybe we can credit the charges back and retain them as a customer.

### Comparing the Random Forrest vs Decision Tree

From a predictive power (higher sensitivity), Random Forrest is a better model. However the difference in Sensitivity between the Pruned Decision tree and Random Forest was negligible. Random Forrest provides variable importance, however we prefer the interpretability of the Decision Tree as it revealed interesting and actionable behavioral pattern in our data.

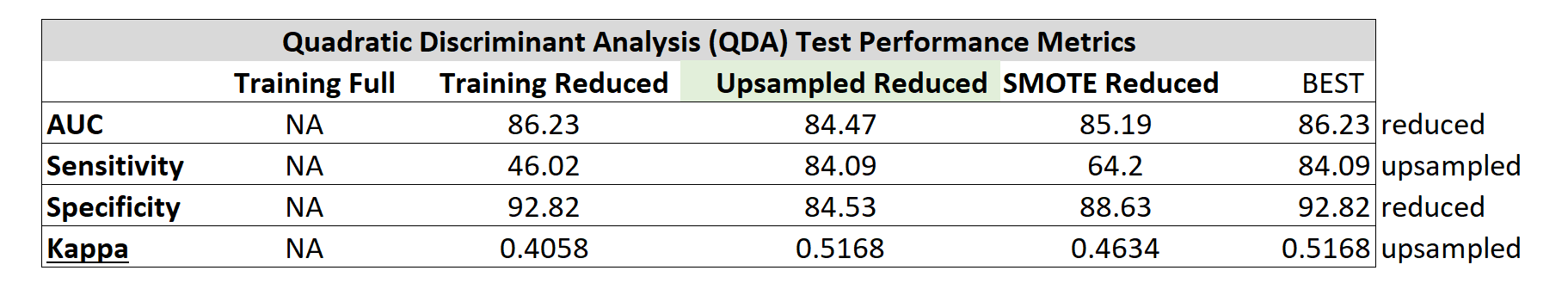
## Model 4: Quadratic Discriminant Analysis (QDA)

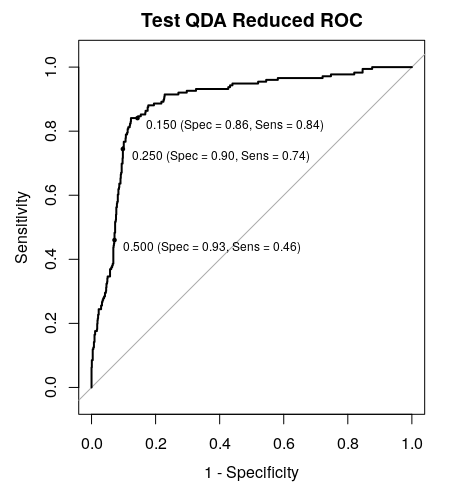
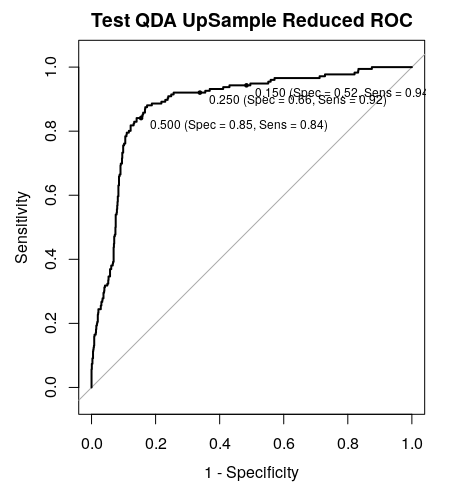
### Predictive Power

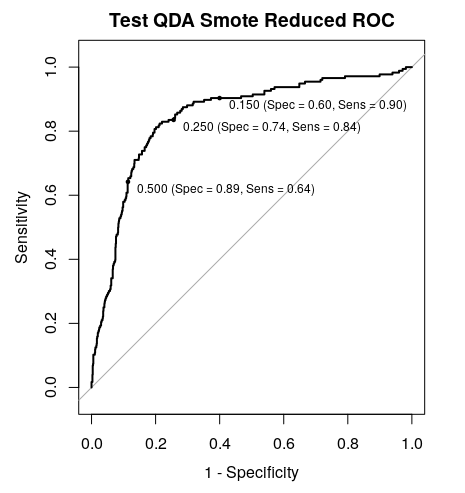
The QDA model did not produce an output for the full training set because of collinearities in the data. For that reason, we created and used the reduced dataset.

For QDA, Sensitivity was significantly lower on the reduced dataset. The upsamling and SMOTE techniques improved both sensitivity and kappa, with the upsampling having the higher metrics.

The best performance was on the upsample reduced dataset.







### Interpretability

QDA only outputs the group means, the averages of each predictor within each class. It is not going to give us with the specific variable importance or coefficient loadings. Thus, we don’t find this model to be interpretable.

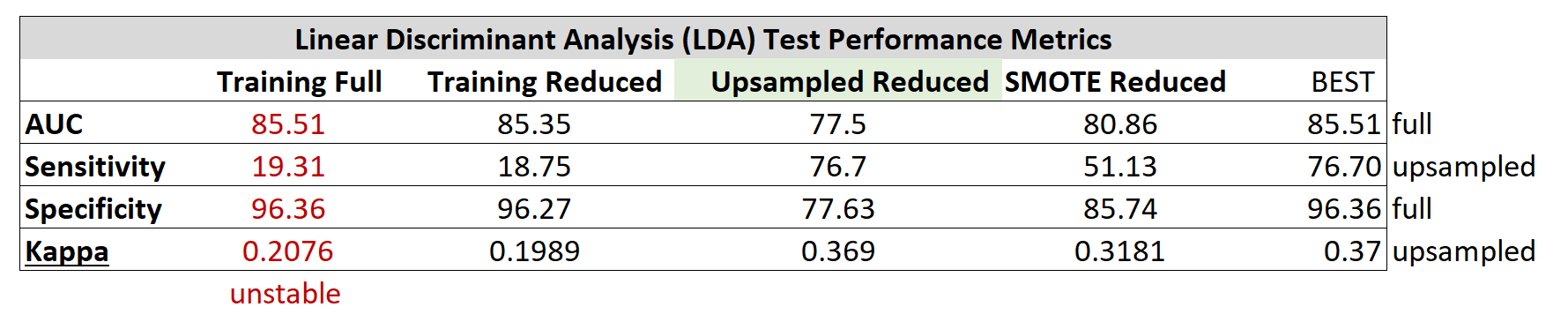
## Model 5: Liner Discriminant Analysis (LDA)

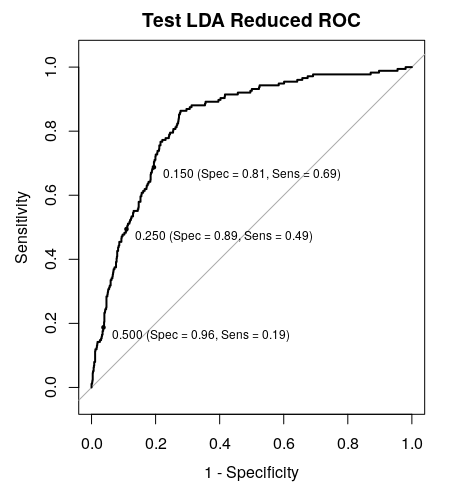
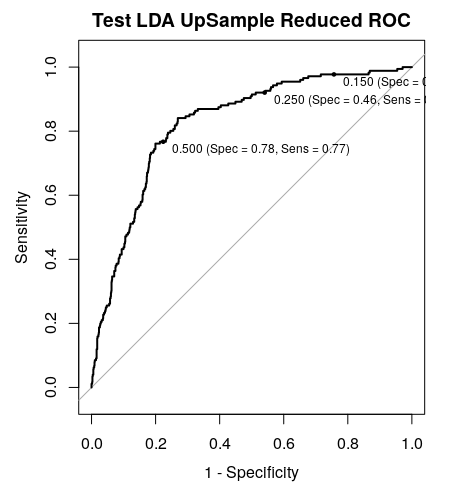
### Predictive Power

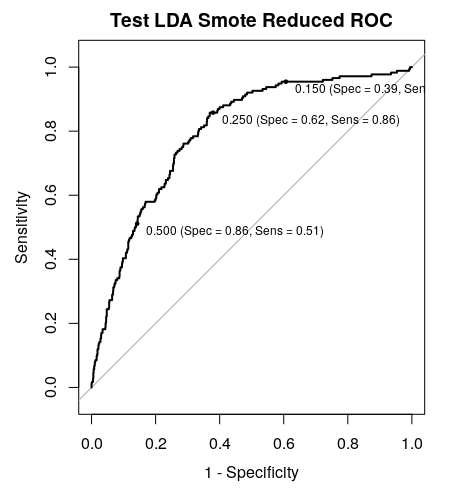
The LDA model produced a rank deficient error for the training full dataset. It still provided parameter estimates and the results, however these results are unstable and should not be trusted. It produced the best AUC and Specificity, and this is another argument why we should not fit linear models when we have collinearity in the data and why we should not use AUC and Specificity as deciding factors to choose the best model.

The reduced dataset gave the worst Sensitivity of 18.75 and Kappa of 0.1989. However, the Upsampling and SMOTE techniques seem to have produced good test results. Specifically, Upsampled produced the highest Sensitivity of 76.7 and highest kappa of 0.369.

For the LDA set of models we would choose the best model performance on the Upsampled dataset,



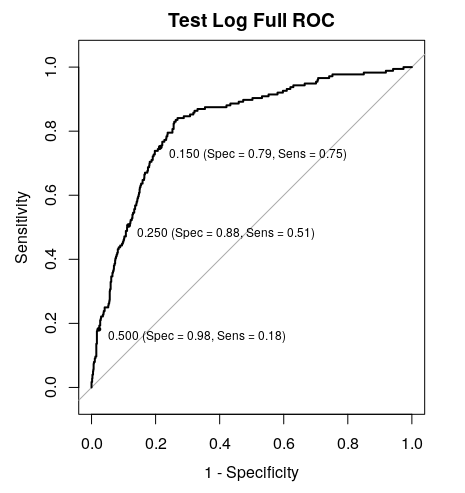
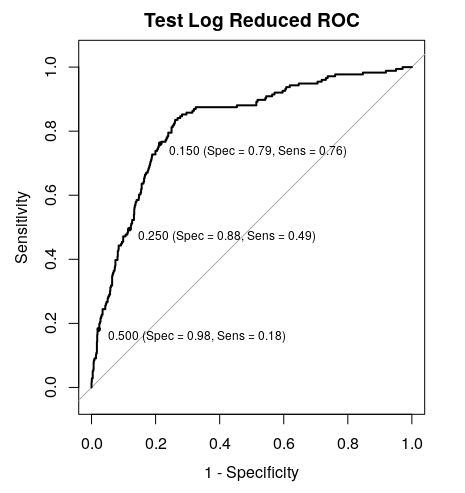


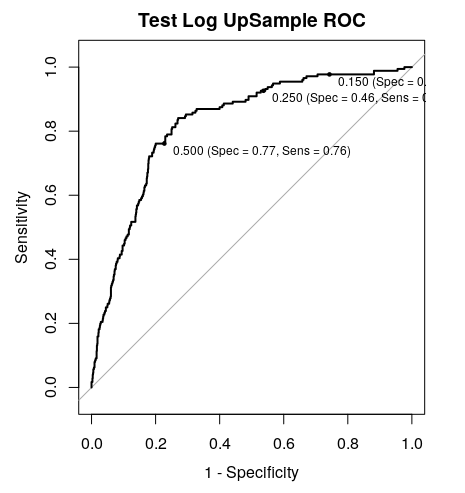
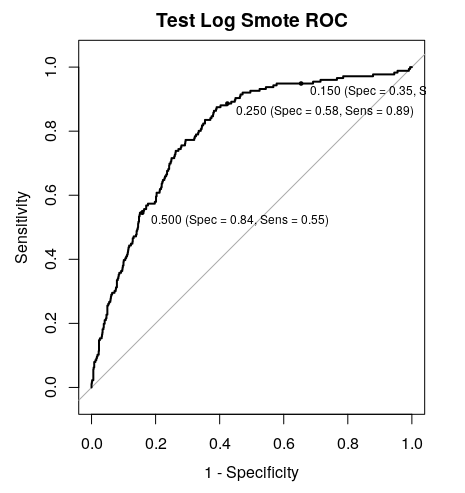


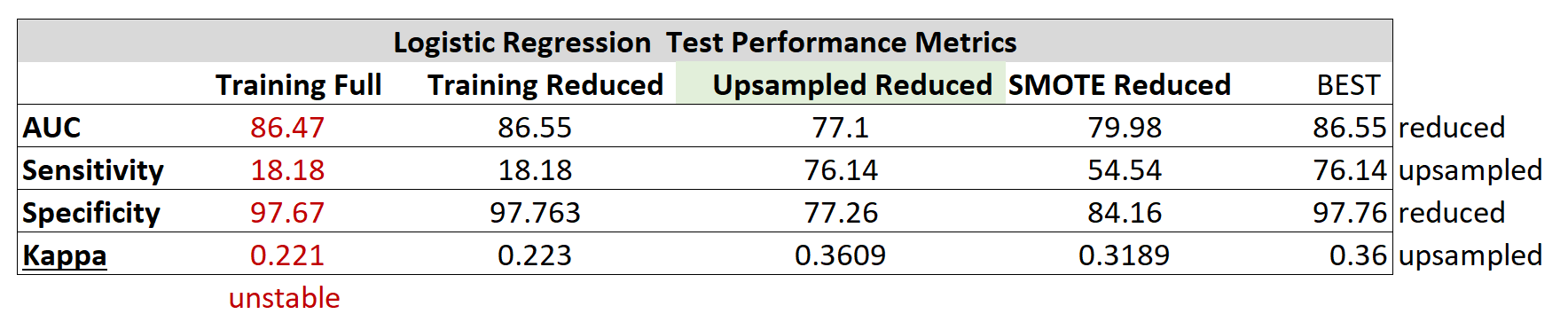
## Model 5: Logistic Regression

### Predictive Power

The logistic regression models also provided an error for the full training set. It provided estimates however these can be unstable, thus should not be used.

The reduced dataset showed very poor performance in identifying the churners as measured with Specificity and kappa. Smote showed improvement in both Sensitivity and Specificity, but the Upsampled training dataset showed the best test metrics.

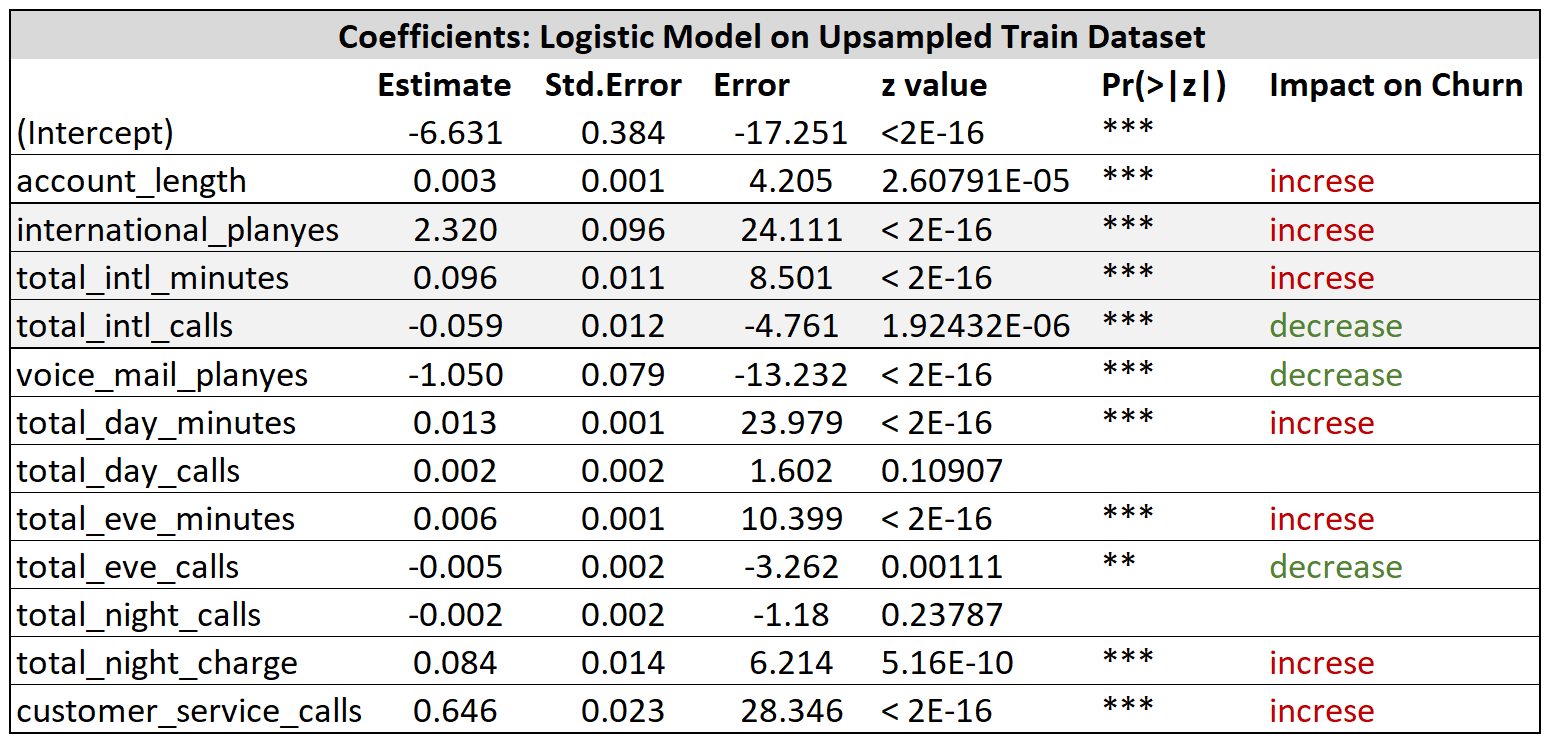




### Interpretability

From interpretability perspective the Logistic model is a clear winner. We can tell not only what is significant in the model, but also understand the impact of churn whether there is a positive or relationship between a the variable and the log probabilities to churn.

The most interesting aspect is something that we saw in the decision trees as well. Customers who have international plans are more likely to churn, but if they make more calls they are less likely to churn. This provides us with an actionable insight, maybe we should target customers who have an international plan but are not utilizing it.



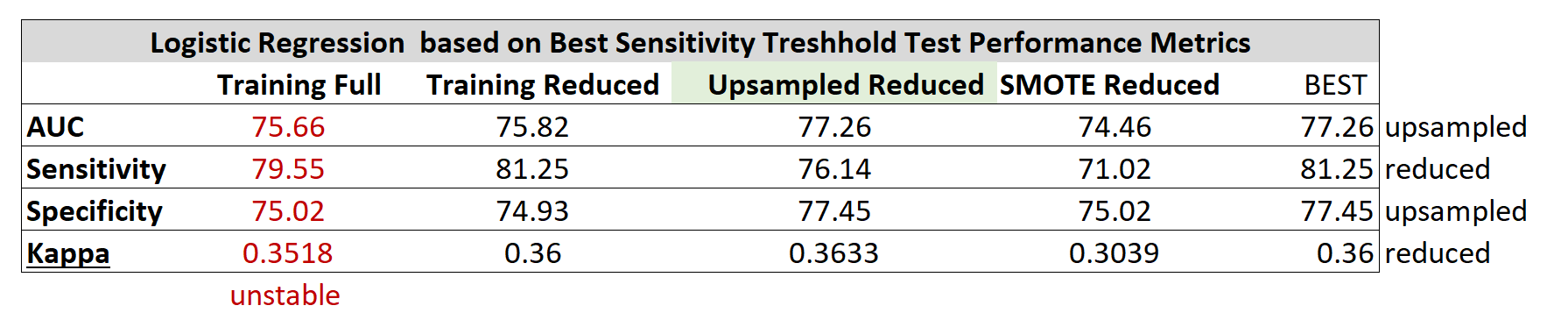
## Model 6: Logistic Regression based on Youden’s J Index

### Predictive Power

Lastly, we fit a logistic regression and for each training set we calculated the best cutoff probability that maximizes sensitivity. We then made took the predictions of the models and applies the best threshold to classify the predictions.



Comparing the results, we can see that the best performance in terms of Sensitivity and Kappa is the reduced dataset. We can see that manually adjusting the threshold from 0.5 to 0.1322 our sensitivity is high and our overall AUC is much lower.



## Interpretability

This model has the same predictions as the first logistic regression we fit, only the probability threshold cutoff is changing. Thus, it will have the same coefficients and interpretability.

# SUMMARY ON CLASSIFICATION MODELS

From the model fitting exercise, we draw the following conclusions:

* Accuracy (Area Under the Curve) is not a good performance metric when we have imbalanced classes, we should use sensitivity and kappa for model selection purposes
* For linear models:
  + running the models with a default threshold of 0.5 results in poor sensitivity
  + Sensitivity increased when upsampling and SMOTE techniques were used
  + Collinearity produced unstable parameter estimates
* Removing highly correlated variables based on pairwise correlations can result in loss of predictive power (total charge variable)
* Random Forrest performed very well regardless of the imbalance of the classes

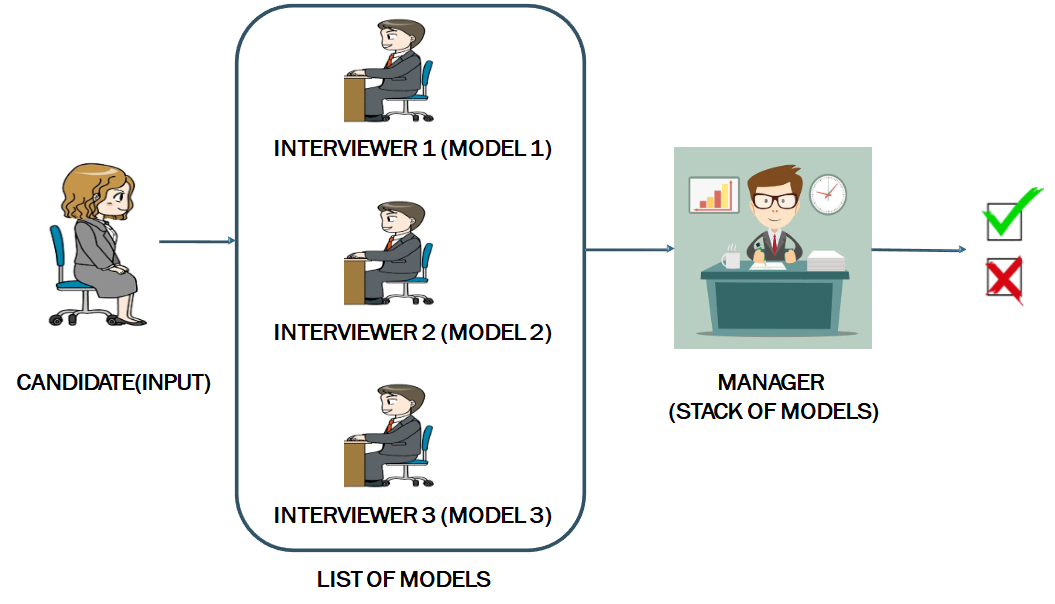
# Ensemble Models

caretEnsemble is a package for making ensembles of caret models.

caretEnsemble has 3 primary functions: **caretList**, **caretEnsemble** and **caretStack**.

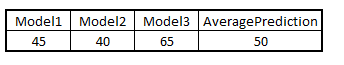
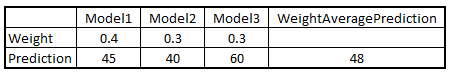
* caretList is used to build lists of caret models on the same training data, with the same re-sampling parameters.
* caretEnsemble and caretStack are used to create ensemble models from such lists of caret models.
* caretEnsemble uses a glm to create a simple linear blend of models
* caretStack uses a caret model to combine the outputs from several component caret models.

Consider an example of a candidate going through multiple rounds of job interviews. The final decision of candidate’s ability is generally taken based on the feedback of all the interviewers. Although a single interviewer might not be able to test the candidate for each required skills and traits, the combined feedback of multiple interviewers usually helps in better assessment of the candidate.



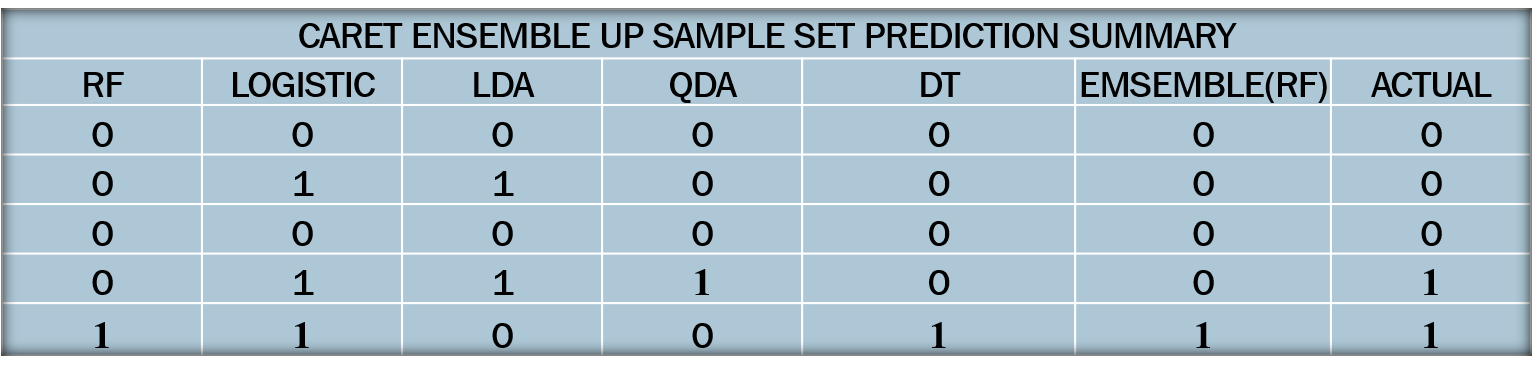
### Types of ensembling

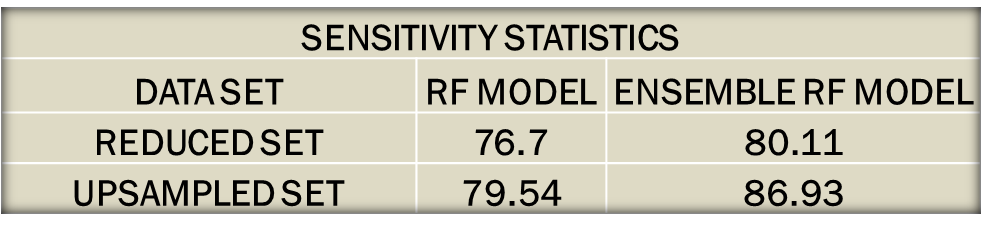
Some of the basic concepts which you should be aware of before we go into further detail are:

* **Averaging:**It’s defined as taking the average ofpredictions from models in case of regression problem or while predicting probabilities for the classification problem.[](https://s3-ap-south-1.amazonaws.com/av-blog-media/wp-content/uploads/2017/02/14160427/Average.png)
* **Majority vote:**It’sdefined astaking the prediction with maximum vote / recommendation from multiple models predictions while predicting the outcomes of a classification problem.[https://s3-ap-south-1.amazonaws.com/av-blog-media/wp-content/uploads/2017/02/14160531/voting.png](https://s3-ap-south-1.amazonaws.com/av-blog-media/wp-content/uploads/2017/02/14160531/voting.png)
* **Weighted average:**In this, different weights are applied to predictions from multiple models then taking the average which means giving high or low importance to specific model output.[](https://s3-ap-south-1.amazonaws.com/av-blog-media/wp-content/uploads/2017/02/14161019/Wtaverage1.png)

### Results of Ensembling

Ensembling did improve the sensitivity of the models. We used Random Forrest to combine the predictions from the different models, as input variables for the Ensembling prediction.

****

****

### Advantages and Disadvantages of ensembling

**Advantages**

* Ensembling is a proven method for improving the accuracy of the model
* It is the key ingredient for winning almost all the machine learning hackathons.
* Ensembling makes the model more robust and stable thus ensuring decent performance on the test cases in most scenarios.
* You can use ensembling to capture linear and simple as well non-linear complex relationships in the data. This can be done by using two different models and forming an ensemble of two.

**Disadvantages**

* Ensembling reduces the model interpretability and makes it very difficult to draw any crucial business insights at the end.
* It is time-consuming and thus might not be the best idea for real-time applications.
* The selection of models for creating an ensemble is an art which is really hard to master.

1. <http://www.oliverwyman.com/our-expertise/insights/2013/jun/the-telecom-business-model-at-risk.html> [↑](#footnote-ref-1)
2. <https://www.mckinsey.com/industries/telecommunications/our-insights/gene-therapy-for-telecom-operators-an-interview-with-jon-fredrik-baksaas> [↑](#footnote-ref-2)
3. <https://en.wikipedia.org/wiki/Accuracy_paradox> [↑](#footnote-ref-3)
4. <https://topepo.github.io/caret/variable-importance.html> [↑](#footnote-ref-4)