1 Introduction

For this project, I analyzed a dataset on cell phone plan cancellations. The 28 predictors are of mixed type. The response variable is a binary variable.

2 Analysis Overview

I selected the following models for analysis with the following tuning parameters.

- Linear, all terms
- ElasticNet Interactions
- K-Nearest Neighbors - k = [1,100]
- Random Forest
  - mtry = [1,25]
  - ntrees = 25, 50, 75, 100, 125, 200, 500
- Gradient Boosting Method
  - n.trees = (100, 200, 500, 1000)
  - shrinkage = (0.01, 0.05, 0.25, 0.5)
  - interaction.depth = (1, 3, 5, 10)
  - n.minobsinnode = (2, 4, 8, 16)
- Support Vector Machine
  - Cost = [0.01, 1] in steps of 0.005.

3 Model Selection

I used the repeated cross-validation approach for all the models using the trainControl() object. I used five folds and three repeats for the purposes of cross-validation. To perform cross-validation, I initially used 70% of the data. This was done because these models output a probability, which needs to be converted into a threshold.

For the random forest model, the mtry was tuned in the training routine. I also considered the effect of varying the number of trees. The results are shown in Fig. 1. For 200 or 500 trees, the random forest algorithm appeared to show stable behavior. Random forest attempts with less than 200 trees by contrast showed unstable behavior, with log-loss showing fluctuations with the mtry parameter. I therefore decided to move forward with tuning the random forest model using 500 trees. The results of the model selection competition are shown in Table 1. It can be seen that the linear, ElasticNet, and

Table 1: Cross-validation results for the models under consideration.

Model	**cvLogLoss**	**cvBrier**	**cvROC-AUC**	**cvPR-AUC**
Linear	0.3890	0.1189	0.8915	0.9311
ElasticNet w/interactions	0.3706	0.1123	0.5	0
K-Nearest Neighbors	0.4751	0.1473	0.6782	0.124
Random Forest	0.3034	0.0885	0.9308	0.9167
Gradient Boosting Method	0.2751	0.0811	0.8459	0.890
SVM	0.3173	0.0953	0.9189	0.9477

k-nearest neighbors (KNN) models all performed poorly compared to the boosting methods and the support vector machine (SVM) method. Since the first three models (linear, ElasticNet, and KNN) were all comparatively worse than the remaining three, I moved forward with the remaining three models for further analysis.

I computed model statistics on the training dataset for the random forest and the gradient boosting method. The log-loss on the training set for the random forest model was 0.0787 and the training error on the gradient boosting method was 0.111. This indicates that the random forest model was slightly overfitting the data as compared to the gradient boosting method.

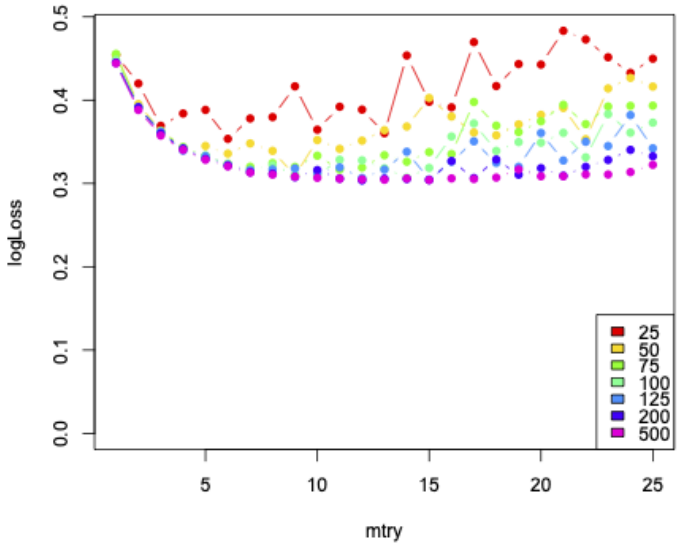

Figure 1: Log-Loss vs. mtry for various numbers of trees.

Given that GBM displayed the lowest cross-validaiton log-loss and also didn't appear to be overfitting as much as random forest from looking at the train data, I selected GBM as the final model. In applying the random forest and GBM models to the remaining 30% of the dataset, I obtained log-losses of 0.0941 and 0.0551, respectively. This further indicates that GBM was a good choise of model. I re-ran a cross-validation using the entire I proceeded to compute the predicted probabilities from the GBM model. To conver probabilities into responses, I computed the threshold that maximized the G-mean value for the held out portion of the dataset (30% of the data). With a threshold value of 0.38, the accuracy was near 99%.

I re-ran a cross-validation using the entire dataset, the resuls of which are shown in Table 2. All models showed an improvement in log-loss, but GBM remained the winner. The predicted probabilities were re-calculated, and the optimal threshold was found to be 0.43, with perfect accuracy as compared to the responses in the dataset.

Table 2: Cross-validation results for the models under consideration using entire dataset.

Model	**cvLogLoss**	**cvBrier**	**cvROC-AUC**	**cvPR-AUC**
Linear	0.3699	0.1142	0.9009	0.9402
ElasticNet w/interactions	0.3508	0.1066	0.5	0
K-Nearest Neighbors	0.4648	0.1473	0.6878	0.1144
Random Forest	0.2755	0.0798	0.9395	0.8931
Gradient Boosting Method	0.2480	0.0726	0.8661	0.1073
SVM	0.3093	0.0924	0.9223	0.9523

4 Final Model

The final model is GBM with the following tuned parameters:

- n.trees = 1000
- interaction depth = 10
- shrinkage = 0.01
- n.minobsinnode = 2