Final Project

Matt Boundy and Jasper Lemberg

Submission Date: 5/7/2021

---

1. Abstract

Using R and a number of methods learned this past semester, we have implemented one random forest model for supervised learning, one k-means model for unsupervised learning, the Boruta Algorithm for feature selection, and grid search to improve on the aforementioned random forest model. These models and methods were implemented on the telecom.data data set, which consists of almost 6,000 surveyed telecom customers on various factors such as gender, whether they have phone service, and monthly costs in order to predict whether they would leave their service behind or not (also known as churn). Using the aforementioned models, we were able to predict which customers were most likely to leave using a number of these factors and then attempted to find ways to improve on these models with varying levels of success. In the end, our models can predict expected churn with an accuracy of around 80%.

---

2. Introduction

Background and Goals

The data set that we are using is from a telecommunications company observing their customer data. There are nearly 6,000 rows and 22 columns. These columns include customer data on individual demographics as well as the services they use, the duration of using the operator's services, the method of payment, and the amount of payment.

The goal of this dataset is to predict the Churn column with a model. Churn is whether or not a customer renews their contract at the end of the year. If you would be able to predict the likelihood of churning for certain customer's you would be able to better target them with deals or promotions so that they would be more likely to renew their contract. Then with customers that would likely remain, less efforts would need to be made to get them to stick with your company. To predict this we will be looking at demographics and contract plans for a variety of customers.

Dataset

Before we could start creating any sort of model, the data had to be cleaned and preprocessed. First, the first two columns of the data set were just unique customer IDs (essentially labels), so they were removed so that they would not affect the final models. From there, all of the variables were turned into numeric types to standardize the data. Because most, if not all, of the predictors were binary, this was fairly simple. Any binary variables (partner, dependents, etc.) were turned into 0s and 1s, with 0 being "No" and 1 being "Yes". For variables that included values like "No phones service" (PhoneService) that relied on previous yes/no variables, those values became -1 if they fell into that category. Lastly, only TotalCharges contained any NA values, which were replaced with the mean of that variable's other values.

For the unsupervised methods, the data was scaled so that the mean was 0 and the standard deviation was 1.

Most of the data lies between 0 and 1 except for tenure, InternetService, Contract, PaymentMethod, MonthlyCharges, and TotalCharges, with some exceptions and outliers. First, the values for SeniorCitizen and PhoneService are skewed right and skewed left respectively, so any points that are 1 for the former and 0 for the latter are outliers. Also, seventy-five percent of the data for InternetService lies above 1, seventy-five percent of the data for Contract lies above 2, and seventy-five percent of the data for PaymentMethod lies below 4. The data for tenure, MonthlyCharges, and TotalCharges is fairly even with means of 32.47, 64.8, and 2298.1 respectively. That being said, there is a fairly large upper bound on the data for TotalCharges because the maximum is 8684.8, whereas the third quartile is only 3841.5.

---

3. Methodology

a. Supervised Method

After considering multiple methods for supervised learning, we decided to use a random forest model for classifying customers, with Churn as a response variable. Before understanding what a random forest is, it is important to know what a decision tree is. In short, a tree is a method of predicting the response value of the data through various rules (James et al. 304). A greedy algorithm based on variable importance determines these rules. For a classification tree, each node of the tree equals a classification (in this case, either 1 for positive churn or 0 for negative churn). This will split the data into segments rather than having a single line split the data. An expansion on that tree idea is the random forest, which consists of multiple bootstrapped trees, each with a random number of predictors (James et al. 319). This method decorrelates the trees and decreases variance on the whole.

However, this method has a number of hyperparameters that we need to tune. To find the right number of trees and predictors, we implemented a five fold cross-validated grid search to find the optimal number of parameters that maximized model accuracy and f1 score.

b. Unsupervised Method

For unsupervised learning, we decided to use a k-means model to cluster the data. The k-means method splits the data into k clusters in order to limit within-cluster variation (James et al. 386). By default, we used squared Euclidean distance to calculate the within-cluster variation. Because there are almost infinitely many options for how to cluster data, the algorithm we used for k-means clustering first assigns random cluster values to all of the data and then recomputes the clusters so that all points inside them are closest to its centroid and only its centroid.

c. Feature Selection

Of course, using all predictors in the model would be unfeasible and could include redundant predictors that have no effect on the final model while also adding computing time. To solve this, we used the Boruta Algorithm to select only the most important features for the model. This algorithm is considered a wrapper class in that it searches through a number of possible feature subspaces and accounts for interactions between features as well (Bontempi 322). More specifically, the Boruta Algorithm randomly compares features' original importance with their importance in random permutations (Kursa). By default, the package takes in a random forest model and, after a top-down search, outputs a graph of how important each feature is to the overall model.

d. Model Improvement

In order to improve our random forest model for supervised learning, we implemented a grid search to find the optimal hyperparameters for the model. Using the caret library, we could search through a number of possible values for the hyperparameters of the random forest model to find one that is most accurate.

---

4. Analysis/Summary

a. Supervised Method: Random Forest

Table 1: Random Forest Confusion Matrix Data

Accuracy	Recall	Precision	F1 Score
0.8032552	0.4839722	0.6827296	0.5656747

As shown above, the accuracy of the model was 0.8032552, the recall was 0.4839722, the precision was 0.6827296, and the F1 score was 0.5656747. While these are not perfect values, These are perfectly feasible results and are much better than the same values for other classification models (bagging, KNN, LDA, etc.).

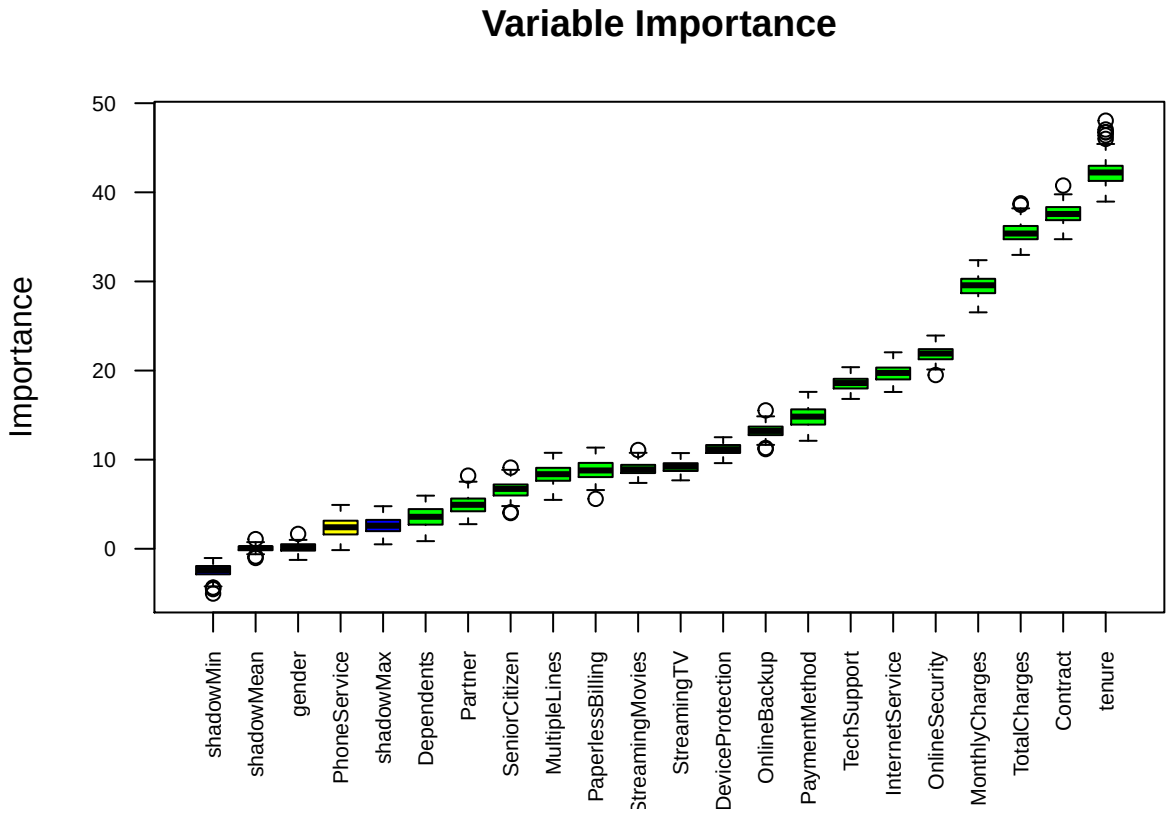
b. Unsupervised Method: K-Means

Table 2: K-Means Confusion Matrix (accuracy = 0.5045105)

	Predicted 0	Predicted 1
Actual 0	1167	1826
Actual 1	1140	1853

Unlike the supervised learning method, the k-means method used to cluster results into 2 clusters was less effective, having an accuracy of just over 0.5. However, this was much better than other methods like hierarchical clustering, which had a similar accuracy, but only because all of the points were lumped into one cluster. Thus, it was the eqivalent of a broken clock being right twice a day.

c. Feature Selection



For the feature selection method we chose to go with the Boruta method. It was able to rank our variables in how important they are in determining the predicted Churn either a 1 or a 0. Using this method we were able to see that tenure, Contract, TotalCharges, MonthlyCharges, and OnlineSecurity were the most important predictor variables. tenure being the most important with around a 50% importance. However, a model consisting of just these five predictors did not perform as well as the model with all predictors, as shown by the table below.

Table 3: Random Forest Confusion Matrix Data with Feature Selection

	Accuracy	Recall	Precision	F1 Score
Before Feature Selection	0.8032552	0.4839722	0.6827296	0.5656747
After Feature Selection	0.7918184	0.4911178	0.6408526	0.5547394

d. Model Improvement

For the method to improve the models, we used the grid search method to improve upon the random forests model that was used in as our supervised method. Although we had used a cross-validated grid search initially to find the optimal values needed for the hyperparameters, this was mostly done with a crude eyeballing method rather than a proper package. Using grid search and running multiple models with the caret library, we were able to find that the optimal mtry for randomforests was 3 rather than 2 which improved our accuracy of the model by 0.1.

5. Discussion and Critics

When applying the random forest model, it is necessary to input the response variable (in this case, Churn) as a factor. Otherwise, it will not be read correctly. Besides that, the only difficulty was the time it took for grid search and 5-fold cross validation, to which the only recommendation is to leave R running in the background and watch a movie with a nice big bowl of popcorn to pass the time.

When applying k-means clustering, we discovered that the data was not clustering well and the model produced a low accuracy (around 50%). Unfortunately, even after scaling the data (something that was necessary considering how spread out some of the points were), the data on the whole was already so dense that it was hard to distinguish individual points. Thus, it is hard to say whether the model could be improved or not.

When applying the Boruta model we discovered the data needed to be more thoroughly cleaned as there were some issues with specific predictor variable types running through the model. We were able to change this by converting certain values to binary to represent boolean statements, as well as change up any data that is missing.

When applying the grid search method we found that there was an issue running our models as the predictor variable was not being read correctly. To combat this we used the as.factor() command to change Churn into a factor so that it could more easily run through the models.

6. Conclusion

For the telecom.data data set, we attempted both supervised and unsupervised learning, with varying effectiveness. Whereas a random forest model for supervised learning had an accuracy of around 80%, a k-means unsupervised model was only 50% accurate, mostly because of how dense the data was. We found that when it comes to the prediction of the Churn for existing customers in a telecommunications company, the best predictors to use are tenure, Contract, MonthlyCharges, OnlineSecurity, and InternetService. Through our use of models we were able to accurately predict the Churn value approximately 80% of the time which still leaves room for improvement. When dealing with some NA values, rather than scrubbing all of the data associated with the values we implemented the mean of the column as the value indicated. This could be a source of error and if we were to revisit this project we may choose to go another direction with how we handle missing data.

References

Bontempi, Gianluca. (2021). “Statistical foundations of machine learning” (2nd edition) handbook.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013.

Miron B. Kursa, Witold R. Rudnicki (2010). Feature Selection with the Boruta Package. Journal of Statistical Software, 36(11), 1-13. URL http://www.jstatsoft.org/v36/i11/.

Zosimov, Radmir. “Telecom Users Dataset.” Kaggle, 22 Feb. 2021, www.kaggle.com/radmirzosimov/telecom-users-dataset.