**Part I: Research Question**

“Disregarding Customer Demographics, what are the top 3 features that have the most impact on Customer Churn at our Telecom Company?”

Goals of Data Analysis:

1. Identify the top 3 features with the most impact. In my experience as an Analyst in Healthcare for a For-Profit Health System, it is better to provide a more granular approach and identify 3-5 “Low-Hanging Fruit” to focus our efforts on. These 3-5 features should be features that have the most impact on our desired outcome.
2. The final model should have an AUC above at least 90% and Accuracy on the unseen test data above at least 85%.

**Part II: Method Justification**

I chose the Random Forest classification algorithm to analyze this dataset. The Random Forest algorithm is based on the same basic logic as Decision Trees where the trees branch out at the nodes based on the combination of your parameters and the decisions made at each node. However, instead of creating only one decision tree, you can set how many trees you would like to create by using the num.trees parameter. I chose to use the default value which is 500 decision trees. The desired number of decision trees work together as an ensemble. Each decision tree results in a single class prediction. The class predictions are totaled and the one with the most votes from all decision trees becomes the random forest model’s classification prediction.

My random forest classification model takes it a step further by adding in hyperparameter tuning and Cross-Validation. For my hyperparameter tuning, I chose to tune the parameter min.node.size from 1 to 10. For Cross-Validation, I chose to split the training dataset into 5 randomly sampled, equally sized parts. As a result, the random forest classification model is trained on each of the 5 parts of the data and tests each of the 10 min.node.size parameters to obtain the highest Accuracy of the parameters.

Assumptions of a random forest classification:

1. Sampling must be representative of the dataset. Other than this, there are no assumptions of the underlying data for random forest models as it is non-parametric.

Libraries used:

**caret:** This library was used to train and construction the random forest classification model. It was also used to create the confusion matrix when evaluating how accurate our model was on new, unseen test data.

**ranger:** This library was used to construction the random forest classification model. Without this library, we could not use the “ranger” method in caret’s train() function.

**caTools:** This library was used to evaluate and plot our random forest classification model’s AUC.

**Part III: Data Preparation**

Data Preprocessing Goal:

1. One data “preprocessing” goal was to remove all customer demographics from the dataset. With random forest and decision tree models, we do not need to normalize or preprocess the data. The algorithms work well with non-normalized, non-preprocessed data. The data “straight-off-the-shelf”.

|  |  |
| --- | --- |
| **Variable Name** | **Type** |
| Churn | Categorical |
| Outage\_sec\_perweek | Continuous |
| Email | Continuous |
| Contacts | Continuous |
| Yearly\_equip\_failure | Continuous |
| Techie | Categorical |
| Contract | Categorical |
| Port\_modem | Categorical |
| Tablet | Categorical |
| InternetService | Categorical |
| Phone | Categorical |
| Multiple | Categorical |
| OnlineSecurity | Categorical |
| OnlineBackup | Categorical |
| DeviceProtection | Categorical |
| TechSupport | Categorical |
| StreamingTV | Categorical |
| StreamingMovies | Categorical |
| PaperlessBilling | Categorical |
| PaymentMethod | Categorical |
| Tenure | Continuous |
| MonthlyCharge | Continuous |
| Bandwidth\_GB\_Year | Continuous |
| Item1 | Continuous |
| Item2 | Continuous |
| Item3 | Continuous |
| Item4 | Continuous |
| Item5 | Continuous |
| Item6 | Continuous |
| Item7 | Continuous |
| Item8 | Continuous |

Data Prep Steps:

1. Import the raw dataset

2. Remove customer demographics by indexing

3. Set seed for random sampling of data

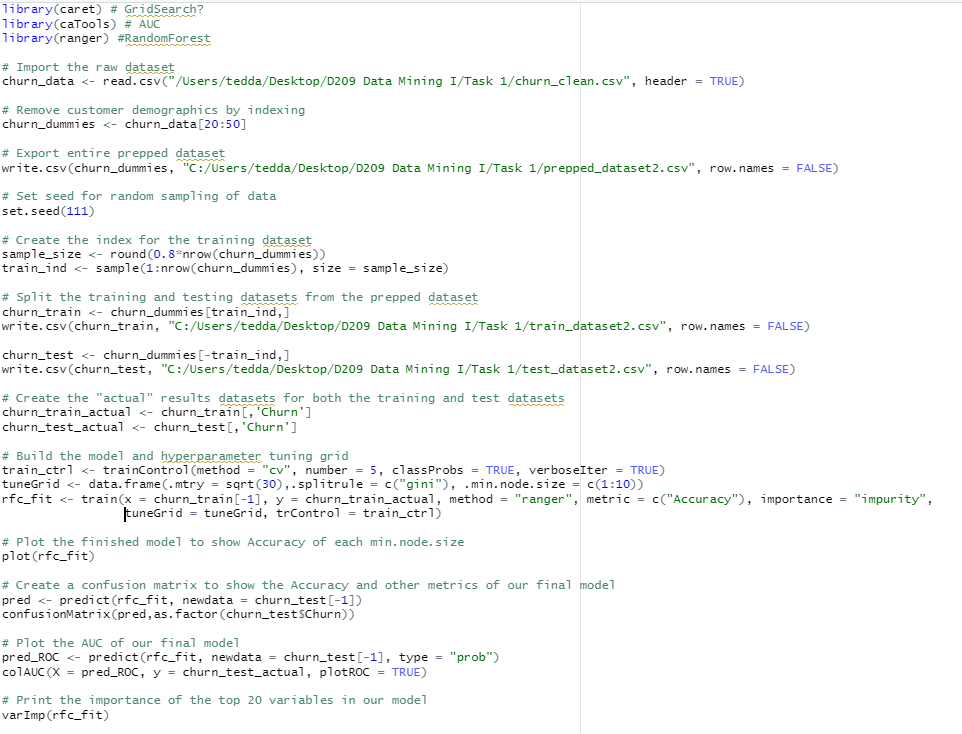
4. Create the index for the random sampling of data

5. Split the data into train and test datasets

6. Create the “actual” results datasets for both the training and test datasets

Please see “Task 2.R” for a script of the annotated code and see below for a screenshot of the same code.

Please see “prepped\_dataset2.csv” for my cleaned, prepped dataset.



**Part IV: Analysis**

Please see “train\_dataset2.csv” and “test\_dataset2.csv” for my split training and test datasets.

My random forest classification model uses Cross-Validation and hyperparameter tuning. For my hyperparameter tuning, I chose to tune the parameter min.node.size from 1 to 10. This parameter is the minimal node size for each node to be split. For Cross-Validation, I chose to split the training dataset into 5 randomly sampled, equally sized parts. As a result, the random forest classification model is trained on each of the 5 parts of the data and tests each of the 10 min.node.size parameters to obtain the highest level of accuracy.

For my other parameters, I used mtry of sqrt(30), which is the number of independent variables, and splitrule of “gini”. The importance was used and set to “impurity” to find each variables level of importance for our random forest classification model. The metric used to evaluate the model was “Accuracy”. In the end, best model had a min.node.size of 8. This can be noted in the plot of rfc\_fit which shows the accuracy of each min.node.size tested.

To evaluate this model against unseen test data, I used the predict() method and created a confusion matrix using caret::confusionMatrix(). The resulting accuracy was 0.892 which is very close to 90%. To further evaluate this model against unseen test data, I plotted the AUC of our final model using caTools::colAUC. The results were an AUC of 0.947 which satisfies my initial goal of above 90%. With 1 being a perfect and always correct model, my random forest classification model is very close. Since this is a classification model and problem, I cannot use MSE to evaluate our model. MSE is used in regression models/problems to evaluate the models.

Please see “Task 2.R” for the annotated code which walks you through each step of the prediction analysis.

**Part V: Data Summary and Implications**

The metric used to evaluate the model was “Accuracy”. In the end, best model had a min.node.size of 8. This can be noted in the plot of rfc\_fit which shows the accuracy of each min.node.size tested. To evaluate this model against unseen test data, I used the predict() method and created a confusion matrix using caret::confusionMatrix(). The resulting accuracy was 0.892 which is very close to 90%. Accuracy shows the percentage of correctly predicted classifications by our model.

To further evaluate this model against unseen test data, I plotted the AUC of our final model using caTools::colAUC. The results were an AUC of 0.947 which satisfies my initial goal of above 90%. Since this is a classification model and problem, I cannot use MSE to evaluate our model. MSE is used in regression models/problems to evaluate the models.

The three variables with the most impact on Churn were Tenure, Bandwidth\_GB\_Year, and MonthlyCharge. This can be viewed by using varImp() on our model. An implication of my model would be to test it on more data in the future studies as there is a difference of about 5% from the accuracy to AUC. This could indicate that the model is slightly overfit and not performing well on new data.

One limitation of my data analysis is that my random forest classification model is currently slow and it would be ineffective in real-time situations where a much larger amount of trees was designated. These results are also only for 1 random seed which can vary depending on the seed set. To minimize these potentially varying results, I used cross-validation to split the data into multiple folds for analysis.

Recommendations:

Based on our top three most impactful variables, provide incentives to customers in tiers based on how long they’ve used our services that reduce the customer’s monthly charge amount by a small, competitive percentage.

Also, based on the tenure tiers and if there is a current limit, reduce any bandwidth\_gb\_year limits of our products.

Please see all PNG files for the code and their outputs.

**Part VI: Demonstration**

Please see my Panopto Presentation.

**I did not use any outside sources.**