Tucker Atwood
WGU MSDA
D209 Task 2
4/17/24

## A1. Research Question

Using data from the "churn" data set, I will answer the question, "Can a random forest regression method be used to predict customer tenure?"

## A2. Analysis Objectives

The goal of this analysis is to understand the relationship between various factors and the amount of time a customer stays with the service. A random forest regression method will be used to predict customer tenure based on several attributes. These predictions may then be used to identify which customers should be targeted for specific actions to get them to stay with the service longer.

## B1. Method Justification

Random forest regression makes predictions based on the average results of several decision trees. A decision tree is a non-parametric regression method that uses the determined relevant variables to break down a data set into smaller, more precise subsets, where each subset contains observations with similar characteristics. At each break, a "decision" is made which splits the data into two or more groups based on one of the relevant variables, choosing a value that ensures the resulting groups differ as much as possible. These decisions ultimately lead to a customer tenure prediction based on the values of the relevant variables.

Random forest regression is an ensemble machine learning method that uses bootstrap aggregation, which is a random sampling of the training set with replacement, to construct many decision trees and average their individual predictions. This creates a prediction that is more accurate and less likely to overfit the data. Each decision tree within a random forest regression chooses a subset of the relevant variables and makes a prediction as described. The mean of all predictions made by the decision trees is then used to construct the random forest regression model, which makes a final prediction of customer tenure based on relevant variables.

The data set will be split into a training set and a test set. The training set will be used to build the random forest model, while the test set will be used to evaluate the model, where the ability to accurately predict customer tenure will be assessed using regression performance metrics. The **expected outcome** of this analysis is to be able to accurately predict customer tenure based on several relevant factors, which can be used to increase the time a customer spends with the service.

## B2. Method Assumption

The random forest regression method assumes that the samples generated from the training set are sufficiently **representative of the overall data set**. Since the method uses bootstrap aggregation to combine the results of several decision trees, the samples used for each decision tree must accurately represent the data. For this analysis, this assumption can be deemed accurate; the training set and test set are created by randomly sampling from the churn data set and the samples in each decision tree within the random forest are randomly generated. It is reasonable to assume the samples generated are representative of the overall data.

## B3. Packages Used

This research question will be answered using the programming language R in the RStudio environment. Within R, the following packages will be utilized:

- **plyr** and **dplyr**: These packages will be used to perform data frame manipulation tasks, such as revaluing observations to make the regression method more accurate.
- **naniar**: This package will be used to check for missing (NA) values within the initial data set. These values must be checked and treated to make the regression results valid.
- **fastDummies**: This package will be used to perform a one-hot encoding process on the necessary categorical data, creating dummy variables for nominal factors. This is essential so that these factors may be included in the regression method.
- **ranger**: This package will be used to conduct the random forest regression method, which is the basis for the analysis.

## C1. Preprocessing Goal

To ensure the regression method will be as accurate and valid as possible, the data must be transformed. The **goal for this transformation is to include only variables relevant to Tenure, and for all of the predictor variables to be expressed as numeric data.**

This goal is relevant to the random forest method because regression is being used to make numeric predictions, so metrics that measure the performance of the model will be more accurate with numeric predictor variables. **All categorical variables must be re-expressed numerically** using 0's and 1's so that decision tree splits are more accurate. Pre-existing numeric variables will not be transformed (i.e. normalized) because numeric variables do not need to be scaled for tree-based methods.

This goal will be addressed in section C3.

## C2. Variables Used

The following variables have been determined to be relevant to the prediction of customer tenure and will be used in the random forest regression method:

- **Churn**, a categorical variable that represents whether or not a customer has left the service (Yes or No).
- **Tenure**, a numeric variable that represents the number of months a customer has been with the service.
- **MonthlyCharge**, a numeric variable that represents the average amount of money charged to the customer's account per month.
- **Age**, a numeric variable that represents the customer's current age.
- **Children**, a numeric variable that represents the number of children the customer has.
- **Bandwidth_GB_Year**, a numeric variable that represents the average data usage for the customer per year, in gigabytes.
- **Techie**, a categorical variable that represents whether or not the customer identifies as technologically inclined (Yes or No).
- **Multiple**, a categorical variable that represents whether or not the customer has more than one phone line (Yes or No).
- **OnlineSecurity**, a categorical variable that represents whether or not the customer has signed up for an online security add-on program (Yes or No).
- **TechSupport**, a categorical variable that represents whether or not the customer has signed up for a technical support add-on program (Yes or No).
- **Contract**, a categorical variable that represents the customer's contract type (Month-to-Month, One-year, or Two-year).
- **InternetService**, a categorical variable that represents the customer's internet service provider (DSL, Fiber Optic, or None).

## C3. Data Preparation

To prepare the data for analysis, the following steps were performed:

- **Data cleaning**
    - **Full duplicates**, defined as observations for which every variable is a match with another observation, were detected. Zero full duplicates were found.
    - **Partial duplicates**, defined as observations for which a subset of variables match another observation, were detected by searching for matches on the "Customer_id" variable. Zero partial duplicates were found.
    - **Missing values**, defined as entries of "NA" in any variable, were detected. Zero missing values were found.
    - **Outliers**, defined as values significantly higher or lower than other established values within the variable, were detected for the relevant numeric variables.

- Zero outliers were found for the variables Tenure, Age, MonthlyCharge, and Bandwidth_GB_Year.
- 401 outliers were found for the variable Children. The range of outliers was 8-10, which was determined to be reasonable and acceptable. All outliers in the Children variable were retained.

The following code executes the data cleaning process as described. An executable version of this code can be found in the attached file: Atwood_D209_Task2_Code.R.

```
sum(duplicated(churn)) # checks for full duplicates (0)

library(plyr) # using plyr package
library(dplyr) # using dplyr package

churn %>%
  count(Customer_id) %>%
  filter(n > 1) # checks for partial duplicates with matching Customer_id (0)

library(naniar) # using naniar package
n_miss(churn) # total missing values (0)

hist(churn$Children) # visualization of Children data; skewed right
boxplot(churn$Children) # 3 outlier values appear
chi_outliers <- churn %>%
  filter((Children < quantile(churn$Children, 0.25, na.rm = TRUE) - IQR(churn$Children, na.rm
= TRUE) * 1.5)
        | (Children > quantile(churn$Children, 0.75, na.rm = TRUE) + IQR(churn$Children, na.rm
= TRUE) * 1.5))
# find outliers using IQR method
count(chi_outliers) # 401 outliers
summary(chi_outliers$Children) # outlier range is 8-10

hist(churn$Age) # visualization of Age data; relatively uniform
boxplot(churn$Age) # no outliers present
age_outliers <- churn %>%
  filter((Age < quantile(churn$Age, 0.25, na.rm = TRUE) - IQR(churn$Age, na.rm = TRUE) * 1.5)
      | (Age > quantile(churn$Age, 0.75, na.rm = TRUE) + IQR(churn$Age, na.rm = TRUE) * 1.5))
# find outliers using IQR method
count(age_outliers) # confirms zero Age outliers
```

```
hist(churn$Tenure) # visualization of Tenure data; bimodal
boxplot(churn$Tenure) # no outliers present
ten_outliers <- churn %>%
  filter((Tenure < quantile(churn$Tenure, 0.25, na.rm = TRUE) - IQR(churn$Tenure, na.rm = TRUE) * 1.5)
         | (Tenure > quantile(churn$Tenure, 0.75, na.rm = TRUE) + IQR(churn$Tenure, na.rm = TRUE) * 1.5))
# find outliers using IQR method
count(ten_outliers) # confirms zero Tenure outliers

hist(churn$MonthlyCharge) # visualization of MonthlyCharge data; normal distribution
boxplot(churn$MonthlyCharge) # no outliers present
mon_outliers <- churn %>%
  mutate(mon_z = scale(churn$MonthlyCharge)) %>%
  filter(mon_z > 3 | mon_z < -3)  # find outliers using z-score method
count(mon_outliers) # confirms zero MonthlyCharge outliers

hist(churn$Bandwidth_GB_Year) # visualization of Bandwidth_GB_Year data; bimodal
boxplot(churn$Bandwidth_GB_Year) # no outliers present
bgy_outliers <- churn %>%
  filter((Bandwidth_GB_Year < quantile(churn$Bandwidth_GB_Year, 0.25, na.rm = TRUE) - IQR(churn$Bandwidth_GB_Year, na.rm = TRUE) * 1.5)
         | (Bandwidth_GB_Year > quantile(churn$Bandwidth_GB_Year, 0.75, na.rm = TRUE) + IQR(churn$Bandwidth_GB_Year, na.rm = TRUE) * 1.5))
# find outliers using IQR method
count(bgy_outliers) # confirms zero Bandwidth_GB_Year outliers
```

- **Re-expression of categorical variables**
    - The variables Churn, Techie, Multiple, OnlineSecurity, and TechSupport were converted to numeric by replacing all "No" responses with "0" and "Yes" responses with "1."
    - The variables Contract and InternetService were converted to numeric by using a one-hot encoding process to create dummy variables for each unique value in each variable. This created six new variables: Contract_One_Year, Contract_Two_Year, Contract_Month_to_Month, InternetService_DSL, InternetService_Fiber_Optic, and InternetService_None.
    - Note: a k-1 method was **not** used to create these dummy variables; the random forest regression method does not require this, since it does not create linear combinations of all independent variables. This information was obtained from: https://www.bzst.com/2015/08/categorical-predictors-how-many-dummies.html

The following code executes the re-expression of categorical variables process as described. An executable version of this code can be found in the attached file: Atwood_D209_Task2_Code.R.

```
churn$Churn <- as.numeric(revalue(churn$Churn, replace = c("No" = 0, "Yes" = 1)))
churn$Techie <- as.numeric(revalue(churn$Techie, replace = c("No" = 0, "Yes" = 1)))
churn$Multiple <- as.numeric(revalue(churn$Multiple, replace = c("No" = 0, "Yes" = 1)))
churn$OnlineSecurity <- as.numeric(revalue(churn$OnlineSecurity, replace = c("No" = 0, "Yes"
= 1)))
churn$TechSupport <- as.numeric(revalue(churn$TechSupport, replace = c("No" = 0, "Yes" = 1)))
churn$DeviceProtection <- as.numeric(revalue(churn$DeviceProtection, replace = c("No" = 0,
"Yes" = 1)))
# re-expresses necessary categorical binary data as numeric, 0 for No, 1 for Yes

library(fastDummies) # using fastDummies package
churn <- churn %>%
  dummy_cols("Contract") %>%
  rename(Contract_One_Year = 'Contract_One year') %>%
  rename(Contract_Two_Year = 'Contract_Two Year') %>%
  rename(Contract_Month_to_Month = 'Contract_Month-to-month')

churn <- churn %>%
  dummy_cols("InternetService") %>%
  rename(InternetService_Fiber_Optic = 'InternetService_Fiber Optic')
# re-expresses necessary categorical non-binary data as numeric using one-hot encoding
```

- **Selection of relevant variables**
    - The variables from the original "churn" data set were reduced to include only those relevant to the random forest regression method. The remaining variables in the prepared data set are:
        - Churn
        - Tenure
        - Age
        - Children
        - MonthlyCharge
        - Bandwidth_GB_Year
        - Techie
        - Multiple
        - OnlineSecurity
        - TechSupport

        - Contract_One_Year
        - Contract_Two_Year
        - Contract_Month_to_Month
        - InternetService_DSL
        - InternetService_Fiber_Optic
        - InternetService_None
    - All of these variables are expressed numerically in the prepared data.

The following code executes the variable selection process as described. An executable version of this code can be found in the attached file: Atwood_D209_Task2_Code.R.

```
churn <- churn %>%
  select(Churn, Tenure, Age, Children, MonthlyCharge, Bandwidth_GB_Year, Techie, Multiple,
OnlineSecurity, TechSupport, DeviceProtection,
        Contract_One_Year, Contract_Two_Year, Contract_Month_to_Month,
        InternetService_DSL, InternetService_Fiber_Optic, InternetService_None)
# selecting only variables that will be used in the analysis
```

**C4. Prepared Data Set**

The resulting cleaned and prepared data set has been written into a CSV file and attached with the following name: churn_209_task2.csv.

**D1. Data Split**

The initial "churn" data set, as well as the referenced prepared data set, contains 10,000 customers and their observed characteristics. To perform a random forest regression method, the data must be split into two subsets: a **training data set**, which will be used to build the random forest model based on values of the observations; and a **test data set**, which will be used to evaluate the performance of the random forest regression model.

A **random 70/30 split** was utilized to create the two subsets, resulting in a training set consisting of 7000 random customers from the churn data set and a test set consisting of the remaining 3000 customers from the churn data set.

The following code executes the data splitting process as described. An executable version of this code can be found in the attached file: Atwood_D209_Task2_Code.R.

```
set.seed(2)
rand_churn <- churn[sample(10000),]
# randomizes churn rows
```

```
churn_train <- rand_churn[1:7000,]
churn_test <- rand_churn[7001:10000,]
# creates 70-30 split for training and test data
```

The resulting training data set has been written into a CSV file and attached with the following name: train_209_task2.csv.

The resulting test data set has been written into a CSV file and attached with the following name: test_209_task2.csv.

## D2. Classification Procedure and Technique

The process of the random forest regression method can be summarized as follows:

- The training set of 7000 randomized observations from the churn data set was used with bootstrap aggregation to construct several decision trees using the relevant factors: Churn, MonthlyCharge, Age, Children, Bandwidth_GB_Year, Techie, Multiple, OnlineSecurity, TechSupport, Contract_One_Year, Contract_Two_Year, Contract_Month_to_Month, InternetService_DSL, InternetService_Fiber_Optic, and InternetService_None.
- **Bootstrap aggregation** allowed random samples from the training set to be taken with replacement for each decision tree. These samples were determined to be representative of the overall data set.
- Each decision tree made a prediction of customer Tenure based on a subset of the relevant variables. The **number of variables considered in each decision tree was 4**, the square root of the total number of independent variables (16). This is the default method for random forest regression and ensures sufficient variety in outcome responses.
- The **target node size for each decision tree was 5**. This is the default value and refers to the minimum number of values within a subset of data to allow a split to happen. If a split would produce a subset with fewer than 5 values, the split does not take place.
- The **number of decision trees was 500**. This is the default value for random forest regression with the ranger package and ensures enough decision trees will be used to make an accurate prediction from their mean.
- The random forest regression model was constructed based on the average results of the 500 decision trees. This model functions as a predictor of customer Tenure for future observations using the same relevant variables.
- The remaining 3000 values in the churn data set were used as the test set. For each of these observations, the random forest regression model used the values of all relevant variables to **make a prediction of customer Tenure**, based on the results from the training set.
- To measure the effectiveness of the model, the **mean squared error (MSE)** was calculated for the test set. This was calculated by subtracting the actual Tenure value for each customer

from their predicted Tenure value, measuring the **residual** of the observation. These residuals were then squared so that they all were represented by positive values, preventing residuals from canceling each other out. Finally, the mean of these squared residuals for all observations was calculated.

- There were no intermediate calculations required to perform this analysis.
- The intermediate calculations to measure the performance of the model are summarized as follows:

**Input:**

```
churn_test$pred <- predict(forest_model, churn_test)$predictions

churn_test %>%
  mutate(residual = pred - Tenure) %>%
  summarize(mse = mean(residual^2))
```

**Output:**

```
> churn_test %>%
+   mutate(residual = pred - Tenure) %>%
+   summarize(mse = mean(residual^2))
       mse
1 6.496159
```

## D3. Classification Code

The following code executes the random forest regression method as described. An executable version of this code can be found in the attached file: Atwood_D209_Task2_Code.R.

```
library(ranger)

forest_model <- ranger(Tenure ~ ., churn_train)

forest_model

churn_test$pred <- predict(forest_model, churn_test)$predictions

churn_test %>%
  mutate(residual = pred - Tenure) %>%
  summarize(rmse = sqrt(mean(residual^2)))
```

## E1. Accuracy and Mean Squared Error

The following summarizes the random forest regression model created with the training set:

```
Call:
 ranger(Tenure ~ ., churn_train)

Type:                             Regression
Number of trees:                  500
Sample size:                      7000
Number of independent variables:  16
Mtry:                             4
Target node size:                 5
Variable importance mode:         none
Splitrule:                        variance
OOB prediction error (MSE):       6.549211
R squared (OOB):                  0.9906168
```

- Note the Mean Squared Error (MSE), also referred to as the Out-of-bag (OOB) prediction error in this output, is approximately **6.55 for the training set.**
- As calculated above, the MSE was approximately **6.50 for the test set.**
- **Mean Squared Error is a measure of the accuracy of the regression model**, with lower MSE values signifying a more accurate model. MSE represents a measure of the difference between actual customer Tenure values and their predicted values using the model.
- For the predicted variable, Tenure, the MSE values were very similar in the training and test sets: 6.55 and 6.50, respectively. This indicates there are no concerning significant differences between the two sets.
- These MSE values are very low with respect to the predicted variable, customer Tenure. This indicates that **the model can predict Tenure with high accuracy**.
- The **R-squared value** is the coefficient of determination, which **measures how well a model fits its data**. It is expressed as a value between 0 and 1, with values closer to 1 indicating a stronger model.
- The **R-squared value for this model was approximately 0.991**. This indicates that the model fits the data very well.

## E2. Results and Implications

The random forest regression model was able to predict customer Tenure with a low Mean Squared Error (approximately 6.50) and a high R-squared value (approximately 0.991). Due to the observed low Mean Squared Error and high R-squared values for the model, **there is strong evidence to suggest this random forest regression model can predict customer Tenure with great accuracy.**

**These results imply that random forest regression should be used to predict and inform customer tenure**. The selected features represent characteristics of customers that can be used to make predictions regarding the amount of time a customer may continue with the service. For current customers with similar features to previous customers who have not stayed with the service for a long period of time, informed decisions can be made to target these specific customers and increase the amount of time they stay with the service.

## E3. Limitations

There are potential **limitations** to the processes and results of this random forest regression:

- Only a subset of variables was selected to be included in the analysis. These variables were selected due to their perceived practical connection to customer Tenure; however, it is possible for one or more other unselected variables in the initial dataset to have a significant impact on customer Tenure. Including other variables in future random forest regressions may increase the overall performance of the method.
- Outliers were detected for one numeric variable, Children. They were determined to be reasonable and acceptable and were therefore retained. However, if these outliers were the result of data entry errors or other data quality issues, this could affect the given results.
- Random forest regression cannot be used to make extrapolated predictions. For future applications of this model which include values of the relevant variables which exist outside the range of the values used to create this model, predictions may not be accurate. It is suggested to create new training and test sets in this situation or to use a different regression method altogether.

## E4. Recommended Course of Action

Based on the given results, it was determined that customer tenure can be usefully predicted using a random forest regression method. It is recommended to use the factors applied to this method, potentially along with other factors if they increase the performance of this model, to determine customers who are at high risk of staying with the service for only a short amount of time. Customers who share common characteristics in these factors with previous customers who have not stayed for long should be targeted specifically in an attempt to increase the time they stay with the service.

Potential actions include, but are not limited to: offering exclusive deals to these at-risk customers, surveying them regarding their satisfaction with the service, and analyzing other services that may offer better incentives to these customers in particular.

## F. Panopto Video

Please refer to the link attached with a Panopto video recording. The video includes an explanation, execution, and output results of the referred code used to perform this analysis.

## G. Third-Party Code References

WGU Courseware was used as a resource to learn the methods, concepts, and functions used to create the codes in this project, including DataCamp course tracks (datacamp.com), Dr. Festus

Elleh's D209 PowerPoint presentation, and the book *Data Science Using Python and R* by Chantal D. Larose and Daniel T. Larose.

## H. Content References

The general understanding and reasoning for not using k-1 groups for dummy variables was obtained with assistance from the following resource:
https://www.bzst.com/2015/08/categorical-predictors-how-many-dummies.html