D209 Data Mining - Task 1

Scott Babcock WGU - MS, Data Analytics

Created: January 13 2025

A1, Research Question

Can customers at risk of churning be predicted using a k-nearest neighbor (KNN) classifier so that the company can take steps to retain their business?

A2, Goals of Analysis

The analysis aims to develop a supervised machine learning model using KNN to predict customers at risk of churning. Accurate predictions with this model will be valuable so the company can take steps to retain customers and mitigate risk.

B1, KNN Classification Method

The k-nearest neighbors classification method is an appropriate method for this analysis. KNN predicts the response variable by looking at the closest specified number of data points, calculating the distance between the points and the target, and voting on the outcome. In other words, KNN assumes that similar things exist close together. Choosing the correct number of data points to look at (k) is essential as too small of a k value can lead to model overfitting, and too large of a k value can lead to underfitting. The expected outcome of the KNN model is to have the response variable in the test data classified based on its nearest neighbors (Harrison, 2018).

B2, KNN Assumption

A key assumption of the k-nearest neighbors method is that similar data points are often close together. Conversely, data points that are further apart are dissimilar. The algorithm measures the distance between these data points and predicts new data based on how its neighbors are classified (Harrison, 2018).

B3, Libraries & Packages

Several packages were used in this analysis to assist with data manipulation or calculations. First, the Dplyr package was used for data manipulation. Dplyr allowed the selection of specific features and the changing of the names of existing features. The Naniar package was used to identify missing values. The miss_var_summary function creates a table with the number of missing values and the percentage of the dataset the missing values make up. The fastDummies package was used to create dummy columns for one hot encoding. The dummy_cols function from the package allowed for the creation of dummy variables in one simple step while also enabling the choice of removing a dummy column if needed and removing the original data. The corrplot package was used to create a correlation plot among the variables. The rsample package was used to create training and testing splits. The initial_split function allows the user to set a proportion of the data used for the split, and enables stratification of the response variable. The caret package was used for the model building and creation of the confusion matrix. Finally, the caTools package was used to calculate the ROC AUC.

```
library(dplyr)
library(naniar)
library(fastDummies)
library(corrplot)
library(rsample)
library(caret)
library(caTools)
```

C1, Data Preprocessing

One data preprocessing step taken in the analysis was to scale the data. Because KNN measures the distance of points from the response variable, it is critical that the data is scaled so it does not give preference to larger numeric values and, thus, has less accurate results. The scale function was used to achieve this across three non-binary numeric variables. The resulting scaled features have a mean of zero and a standard deviation of one (Bobbitt, 2021).

C2, Dataset Variables

The following variables were used in the initial subset for the analysis. The Churn variable was used as the response variable, what is to be predicted. Churn is a categorical variable. The Age variable is numeric and needs to be scaled. The MonthlyCharge variable is numeric and needs to be scaled. The Bandwidth_GB_Year variable is numeric and needs to be scaled. Marital is a categorical variable re-expressed as a numeric with one hot encoding. Gender is a categorical variable re-expressed as a numeric with one hot encoding. Contract is a categorical variable and ended up being re-expressed as numeric with one hot encoding.

C3, Steps for Analysis

The following steps were followed to prepare the data for the analysis.

- The data from the churn_clean CSV file was loaded into the programming environment.
- The data was previewed, and data types were identified.
- The data was checked for duplicate values. No duplicate values were identified.
- The data was checked for missing values. No missing values were identified.
- The subset of data used in the analysis was selected, and a new data frame was created.
- The numeric variables were checked for outliers. No outliers were present.
- Unique values were identified for each categorical variable. There were not too many values for the creation of dummy variables.
- The categorical variables were re-expressed as numeric with one hot encoding.
- The Churn variable was converted to a factor for the analysis, and the structure was checked to ensure the change occurred.
- The non-binary numeric values were centered and scaled for the analysis to reduce bias.
- Summary statistics and standard deviation were viewed on each of the three variables to verify that the scaling occurred. All had a mean of zero and a standard deviation of one.
- A new data frame was created for a correlation plot. The Churn variable was converted to numeric in this data frame to allow for correlation. The goal of this was to check for correlation among the overarching features.
- The cleaned and transformed data was written to a CSV file.

```r
# load churn dataset
churn_df <- read.csv("churn_clean.csv")

# view data structure and types
str(churn_df)
```

```
## 'data.frame':    10000 obs. of  50 variables:
##  $ CaseOrder           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Customer_id         : chr  "K409198" "S120509" "K191035" "D90850" ...
##  $ Interaction         : chr  "aa90260b-4141-4a24-8e36-b04ce1f4f77b" "fb76459f-c047-4a9d-8af9-e0f7d4
```

```
##  $ UID                 : chr  "e885b299883d4f9fb18e39c75155d990" "f2de8bef964785f41a2959829830fb8a"
##  $ City                : chr  "Point Baker" "West Branch" "Yamhill" "Del Mar" ...
##  $ State               : chr  "AK" "MI" "OR" "CA" ...
##  $ County              : chr  "Prince of Wales-Hyder" "Ogemaw" "Yamhill" "San Diego" ...
##  $ Zip                 : int  99927 48661 97148 92014 77461 31030 37847 73109 34771 45237 ...
##  $ Lat                 : num  56.3 44.3 45.4 33 29.4 ...
##  $ Lng                 : num  -133.4 -84.2 -123.2 -117.2 -95.8 ...
##  $ Population          : int  38 10446 3735 13863 11352 17701 2535 23144 17351 20193 ...
##  $ Area                : chr  "Urban" "Urban" "Urban" "Suburban" ...
##  $ TimeZone            : chr  "America/Sitka" "America/Detroit" "America/Los_Angeles" "America/Los_Ar
##  $ Job                 : chr  "Environmental health practitioner" "Programmer, multimedia" "Chief Fin
##  $ Children            : int  0 1 4 1 0 3 0 2 2 1 ...
##  $ Age                 : int  68 27 50 48 83 83 79 30 49 86 ...
##  $ Income              : num  28562 21705 9610 18925 40074 ...
##  $ Marital             : chr  "Widowed" "Married" "Widowed" "Married" ...
##  $ Gender              : chr  "Male" "Female" "Female" "Male" ...
##  $ Churn               : chr  "No" "Yes" "No" "No" ...
##  $ Outage_sec_perweek  : num  7.98 11.7 10.75 14.91 8.15 ...
##  $ Email               : int  10 12 9 15 16 15 10 16 20 18 ...
##  $ Contacts            : int  0 0 0 2 2 3 0 0 2 1 ...
##  $ Yearly_equip_failure: int  1 1 1 0 1 1 1 0 3 0 ...
##  $ Techie              : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Contract            : chr  "One year" "Month-to-month" "Two Year" "Two Year" ...
##  $ Port_modem          : chr  "Yes" "No" "Yes" "No" ...
##  $ Tablet              : chr  "Yes" "Yes" "No" "No" ...
##  $ InternetService     : chr  "Fiber Optic" "Fiber Optic" "DSL" "DSL" ...
##  $ Phone               : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ Multiple            : chr  "No" "Yes" "Yes" "No" ...
##  $ OnlineSecurity      : chr  "Yes" "Yes" "No" "Yes" ...
##  $ OnlineBackup        : chr  "Yes" "No" "No" "No" ...
##  $ DeviceProtection    : chr  "No" "No" "No" "No" ...
##  $ TechSupport         : chr  "No" "No" "No" "No" ...
##  $ StreamingTV         : chr  "No" "Yes" "No" "Yes" ...
##  $ StreamingMovies     : chr  "Yes" "Yes" "Yes" "No" ...
##  $ PaperlessBilling    : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ PaymentMethod       : chr  "Credit Card (automatic)" "Bank Transfer(automatic)" "Credit Card (auto
##  $ Tenure              : num  6.8 1.16 15.75 17.09 1.67 ...
##  $ MonthlyCharge       : num  172 243 160 120 150 ...
##  $ Bandwidth_GB_Year   : num  905 801 2055 2165 271 ...
##  $ Item1               : int  5 3 4 4 4 3 6 2 5 2 ...
##  $ Item2               : int  5 4 4 4 4 3 5 2 4 2 ...
##  $ Item3               : int  5 3 2 4 4 3 6 2 4 2 ...
##  $ Item4               : int  3 3 4 2 3 2 4 5 3 2 ...
##  $ Item5               : int  4 4 4 5 4 4 1 2 4 5 ...
##  $ Item6               : int  4 3 3 4 4 3 5 3 3 2 ...
##  $ Item7               : int  3 4 3 3 4 3 5 4 4 3 ...
##  $ Item8               : int  4 4 3 3 5 3 5 5 4 3 ...
```

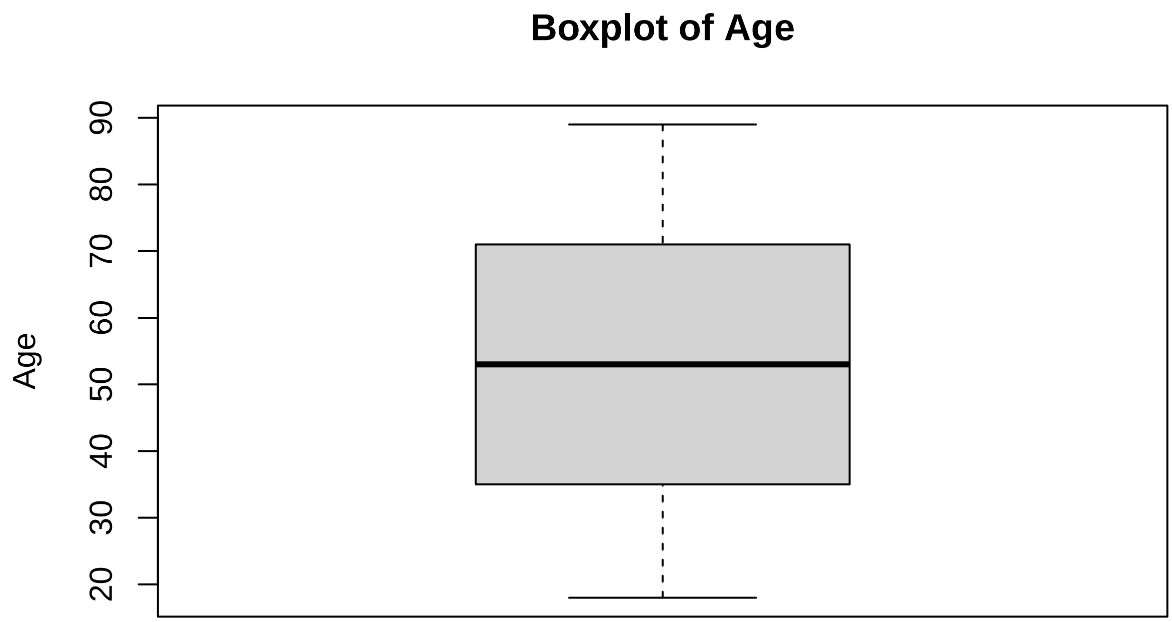
```r
# check for duplicate records [In-text citation:(Getting Started with Duplicates, n.d.)]
sum(duplicated(churn_df))
```

```
## [1] 0
```

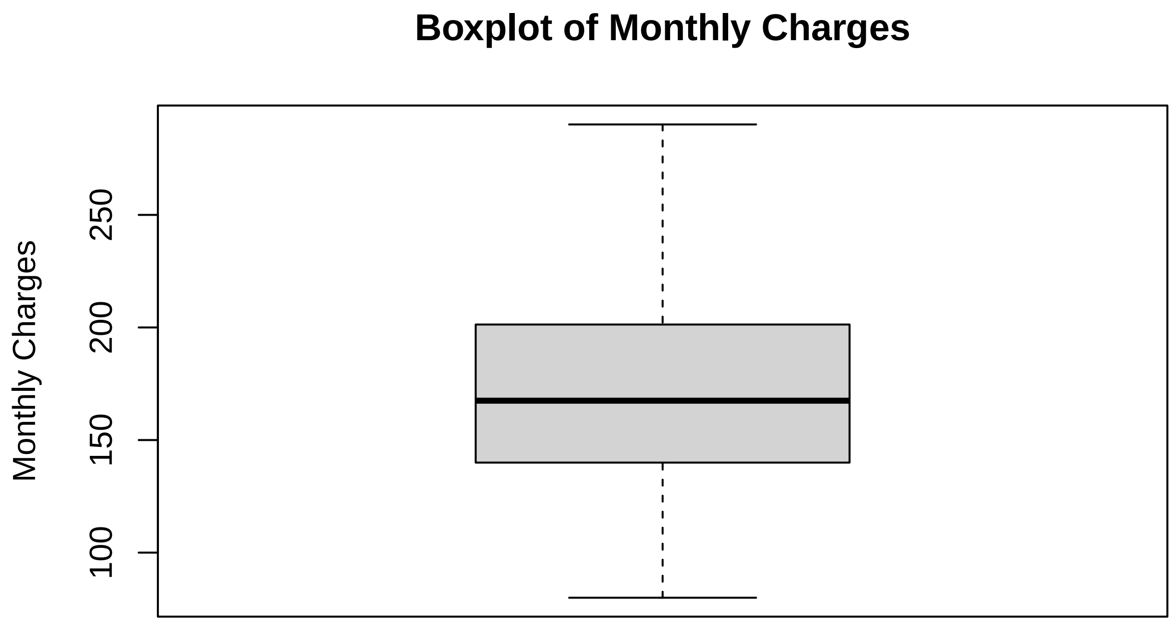
```r
# check for missing values [In-text citation: (Tierney, n.d.)]
miss_var_summary(churn_df)
```

```
## # A tibble: 50 x 3
##    variable    n_miss pct_miss
##    <chr>        <int>    <num>
##  1 CaseOrder        0        0
##  2 Customer_id      0        0
##  3 Interaction      0        0
##  4 UID              0        0
##  5 City             0        0
##  6 State            0        0
##  7 County           0        0
##  8 Zip              0        0
##  9 Lat              0        0
## 10 Lng              0        0
## # i 40 more rows
```

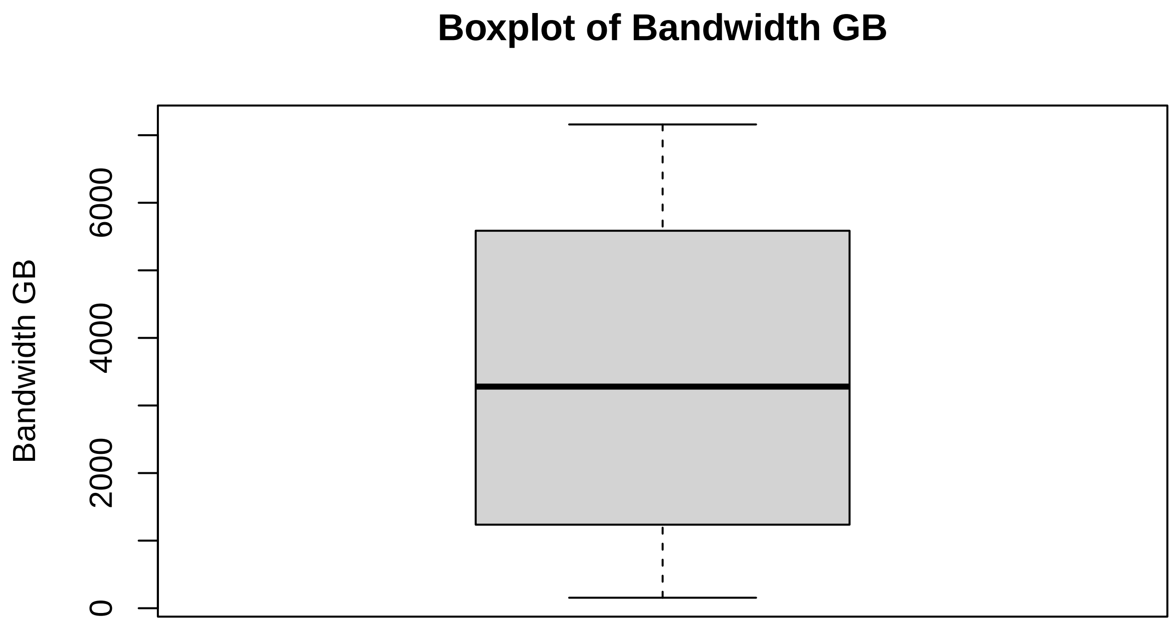
```r
# select variables for KNN model
churn_analysis_initial <- churn_df %>%
  select(
    Churn,
    Age,
    MonthlyCharge,
    Bandwidth_GB_Year,
    Marital,
    Gender,
    Contract
  )

# check for outliers in numeric variables
boxplot(churn_analysis_initial$Age,
        ylab = "Age",
        main = "Boxplot of Age")
```




```r
boxplot(churn_analysis_initial$MonthlyCharge,
        ylab = "Monthly Charges",
        main = "Boxplot of Monthly Charges")
```




```r
boxplot(churn_analysis_initial$Bandwidth_GB_Year,
        ylab = "Bandwidth GB",
        main = "Boxplot of Bandwidth GB")
```




```
# view unique values for each categorical variable
unique(churn_analysis_initial$Churn)
```

```
## [1] "No"  "Yes"
```

```
unique(churn_analysis_initial$Marital)
```

```
## [1] "Widowed"       "Married"       "Separated"     "Never Married"
## [5] "Divorced"
```

```
unique(churn_analysis_initial$Gender)
```

```
## [1] "Male"      "Female"    "Nonbinary"
```

```
unique(churn_analysis_initial$Contract)
```

```
## [1] "One year"       "Month-to-month" "Two Year"
```

```
# transform categorical variables with one hot encoding [In-text citation: (Kaplan, 2020)]
churn_analysis <- dummy_cols(
  churn_analysis_initial,
  select_columns = c("Marital","Gender","Contract"),
```

```
  remove_first_dummy = FALSE,
  remove_selected_columns = TRUE
)

# transform Churn variable to factor
churn_analysis$Churn <- as.factor(churn_analysis$Churn)
str(churn_analysis)
```

```
## 'data.frame':    10000 obs. of  15 variables:
##  $ Churn                  : Factor w/ 2 levels "No","Yes": 1 2 1 1 2 1 2 2 1 1 ...
##  $ Age                    : int  68 27 50 48 83 83 79 30 49 86 ...
##  $ MonthlyCharge          : num  172 243 160 120 150 ...
##  $ Bandwidth_GB_Year      : num  905 801 2055 2165 271 ...
##  $ Marital_Divorced       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Marital_Married        : int  0 1 0 1 0 0 0 1 0 1 ...
##  $ Marital_Never Married  : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Marital_Separated      : int  0 0 0 0 1 0 0 0 1 0 ...
##  $ Marital_Widowed        : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ Gender_Female          : int  0 1 1 0 0 1 0 1 0 1 ...
##  $ Gender_Male            : int  1 0 0 1 1 0 1 0 0 0 ...
##  $ Gender_Nonbinary       : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Contract_Month-to-month: int  0 1 0 0 1 0 1 1 1 0 ...
##  $ Contract_One year      : int  1 0 0 0 0 1 0 0 0 0 ...
##  $ Contract_Two Year      : int  0 0 1 1 0 0 0 0 0 1 ...
```

```
# center and scale non-binary numeric values [In-text citation: (Bobbitt, 2021)]
churn_analysis_scale <- churn_analysis %>%
  mutate(Age = scale(Age),
         MonthlyCharge = scale(MonthlyCharge),
         Bandwidth_GB_Year = scale(Bandwidth_GB_Year)
  )

summary(churn_analysis_scale$Age)
```

```
##        V1
##  Min.   :-1.694700
##  1st Qu.:-0.873400
##  Median :-0.003788
##  Mean   : 0.000000
##  3rd Qu.: 0.865825
##  Max.   : 1.735437
```

```
sd(churn_analysis_scale$Age)
```

```
## [1] 1
```

```
summary(churn_analysis_scale$MonthlyCharge)
```

```
##        V1
##  Min.   :-2.1574
##  1st Qu.:-0.7602
```

```
##   Median :-0.1197
##   Mean   : 0.0000
##   3rd Qu.: 0.6546
##   Max.   : 2.7370
```

```
sd(churn_analysis_scale$MonthlyCharge)
```

```
## [1] 1
```

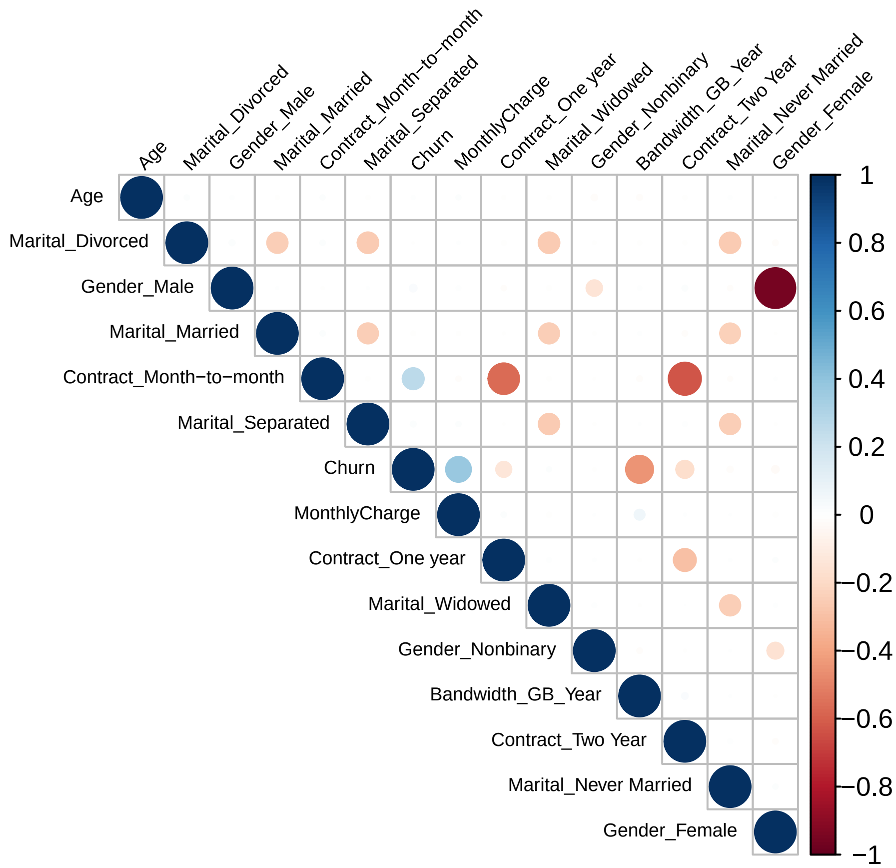
```
summary(churn_analysis_scale$Bandwidth_GB_Year)
```

```
##         V1
##   Min.   :-1.48119
##   1st Qu.:-0.98654
##   Median :-0.05162
##   Mean   : 0.00000
##   3rd Qu.: 1.00389
##   Max.   : 1.72363
```

```
sd(churn_analysis_scale$Bandwidth_GB_Year)
```

```
## [1] 1
```

```
# create correlation plot [In-text citation: (Schork, n.d.)]
cor_data <- churn_analysis_scale %>%
    select( Churn,
            Age,
            MonthlyCharge,
            Bandwidth_GB_Year,
            Marital_Divorced,
            Marital_Married,
            `Marital_Never Married`,
            Marital_Separated,
            Marital_Widowed,
            Gender_Female,
            Gender_Male,
            Gender_Nonbinary,
            `Contract_One year`,
            `Contract_Two Year`,
            `Contract_Month-to-month`)
cor_data$Churn <- as.numeric(cor_data$Churn)

corrplot(cor(cor_data),
         method = "circle",
         tl.cex = 0.6,
         tl.srt = 45,
         tl.col = "black",
         type = "upper",
         order = "hclust")
```



C4, Cleaned Dataset

The cleaned and transformed dataset used in the analysis was written to a CSV file and is included in the submission.

D1, Splitting the Data

The data was split using a 70%/30% ratio for training and testing. Stratification was used on the response variable to ensure the training and testing sets had equal proportions of the Churn variable. The resulting data frames were checked to ensure the stratification ratios were correct.

The training and testing data were written to CSV files and are included in the submission.

```r
# create train/test data with 70/30 split [In-text citation: (Simple Training, n.d.)]
set.seed(444)

split <- initial_split(churn_analysis_scale, prop = 0.7, strata = "Churn")

train <- training(split)
test  <- testing(split)

# ensure equal portions of response variable
sum(ifelse(train$Churn == "Yes",1,0))/nrow(train)
```

```
## [1] 0.264895
```

```r
sum(ifelse(test$Churn == "Yes",1,0))/nrow(test)
```

```
## [1] 0.2652449
```

```r
# write train and test splits to csv
write.csv(train,
          "d209_task1_babcock_train_data.csv",
          row.names = FALSE)

write.csv(test,
          "d209_task1_babcock_test_data.csv",
          row.names = FALSE)
```

D2, Output and Calculations

After all the preprocessing steps, the model was built. The k-nearest neighbors technique was used to analyze the data. Using the caret package, the trainControl function allowed for cross-validation. Cross-validation divides the data into multiple subsets and evaluates the model on each subset. This aims to prevent overfitting (Cross validation, n.d.).

The KNN model was built with Churn as the response variable against the remaining 14 explanatory variables. KNN classifies the Churn variable by looking at the closest specified number of data points, calculating the distance between the points and the target, and voting on the outcome. In this case, the optimal number of neighbors (k) was nine, as it provided the highest ROC AUC value.

The model was then used to make predictions on the unseen test data. After creating the prediction data frame, a confusion matrix was created to assess the model's accuracy on the unseen data. The model produced 85% accuracy, calculated by taking the sum of true positive and negative predictions and dividing them by the total number of observations. This exceeded the no information rate of 73%, which is the rate of the most common class and what could essentially be achieved by random chance. The model's AUC was 0.795, which is respectable. The AUC, or area under the curve, represents a model's ability to discriminate between positive and negative classes (Welcome to D209, n.d.). The AUC scale is between zero and one. An AUC of one would mean the model predicted everything correctly, and an AUC of 0.5 would indicate that the model is only as good as a random chance.

D3, Code Execution

The following code was used to build and train the KNN model.

```r
# train KNN model [In-text citation: (Kuhn, n.d.)]
set.seed(444)

train_control <- trainControl(method = "cv",
                              number = 10,
                              summaryFunction = twoClassSummary,
                              classProbs = TRUE
                              )

knn_fit <- train(Churn ~ .,
```

```
                 data = train,
                 method = "knn",
                 trControl = train_control
                 )

knn_fit
```

```
## k-Nearest Neighbors
##
## 6999 samples
##   14 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 6300, 6299, 6298, 6300, 6300, 6299, ...
## Resampling results across tuning parameters:
##
##   k  ROC        Sens       Spec
##   5  0.8892422  0.9016516  0.6817931
##   7  0.9014457  0.9053417  0.6898750
##   9  0.9089281  0.9103978  0.6850189
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

The code below was used to make predictions using the KNN model on the unseen test data and assess the accuracy and the AUC score.

```
# create predictions with KNN model using test data [In-text citation: (Kuhn, n.d.)]
predictions <- predict(knn_fit, newdata = test) %>%
  bind_cols(test) %>%
  rename_at('...1', ~'Churn_pred')

# create confusion matrix to assess accuracy [In-text citation: (Kuhn, n.d.)]
confusionMatrix(predictions$Churn_pred, predictions$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  2031  262
##        Yes  174  534
##
##                Accuracy : 0.8547
##                  95% CI : (0.8416, 0.8671)
##     No Information Rate : 0.7348
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6136
##
##  Mcnemar's Test P-Value : 3.093e-05
```

```
##
##             Sensitivity : 0.9211
##             Specificity : 0.6709
##          Pos Pred Value : 0.8857
##          Neg Pred Value : 0.7542
##              Prevalence : 0.7348
##          Detection Rate : 0.6768
##    Detection Prevalence : 0.7641
##       Balanced Accuracy : 0.7960
##
##        'Positive' Class : No
##
```

```
# assess ROC AUC [In-text citation: (Kuhn, n.d.)]
colAUC(
  as.numeric(predictions$Churn_pred),
  as.numeric(predictions$Churn)
)
```

```
##                [,1]
## 1 vs. 2 0.7959714
```

E1, Accuracy & ROC AUC

The model produced substantial accuracy and AUC scores. The model produced 85% accuracy, calculated by taking the sum of true positive and negative predictions and dividing them by the total number of observations. This exceeded the no information rate of 73%, which is the rate of the most common class and what could essentially be achieved by random chance. The model's AUC was 0.795, which is respectable. The AUC, or area under the curve, represents a model's ability to discriminate between positive and negative classes (Welcome to D209, n.d.). The AUC scale is between zero and one. An AUC of one would mean the model predicted everything correctly, and an AUC of 0.5 would indicate that the model is only as good as a random chance.

E2, Results & Implications

The analysis aimed to develop a supervised machine learning model using KNN classification to predict customers at risk of churning. This was a successful result based on the model's accuracy and AUC metrics. The model could predict with 85% accuracy, and the AUC was 0.795, both indicators of a strong model. This is undoubtedly a good first step. Additional models could be created with different hyper-tuning parameters to see if the metrics could be improved.

E3, Limitations

One limitation of the data analysis is that it may not have cast a wide enough net regarding the variables selected. The variables used were the optimal variables from a prior analysis. These same variables were used to see if a better model could be produced. A different approach could have been to start with all variables again and use a feature selection method.

E4, Course of Action

The model produced respectable accuracy and AUC metrics. Ideally, a couple more models would be fit with different hyper-tuning parameters to see if the model could be improved. However, as it stands, the company could use the model to make predictions about which customers are at risk of churning. The model would be valuable for the company, as steps can be taken to retain customers and mitigate risk.

F, Panopto Video

A Panopto video recording was created that covered the execution of the code. The video link can be found in the submission.

G, Sources for Code

Bobbitt, Z. (December 10, 2021). How to use the scale() function in R. Statology. Retrieved January 8, 2025, from (https://statology.org/scale-function-in-r/)

Kaplan, J. (November 28, 2020). Making dummy variables with dummy_cols(). fastDummies. Retrieved December 8, 2024, from (https://jacobkap.github.io/fastDummies/articles/making-dummy-variables.html)

Kuhn, M. (n.d.). Machine learning with caret in R [MOOC]. DataCamp. (https://app.datacamp.com/learn/courses/machine-learning-with-caret-in-r)

Schork, J. (n.d.). Correlation matrix in R. Statistics Globe. Retrieved December 10, 2024, from (https://statisticsglobe.com/correlation-matrix-in-r)

Simple Training/Test Set Splitting (n.d.). Rsample. Retrieved January 8, 2025, from (https://rsample.tidymodels.org/reference/initial_split.html)

Tierney, N. (n.d.). Dealing with Missing Data in R [MOOC]. DataCamp. (https://app.datacamp.com/learn/courses/dealing-with-missing-data-in-r)

WGU College of Information Technology (n.d.). Getting Started with Duplicates [PowerPoint slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%2

H, Sources for Content

Bobbitt, Z. (December 10, 2021). How to use the scale() function in R. Statology. Retrieved January 8, 2025, from (https://statology.org/scale-function-in-r/)

Cross validation in machine learning (n.d.). Geeks for Geeks. Retrieved January 13, 2025, from (https://www.geeksforgeeks.org/cross-validation-machine-learning/)

Harrison, O. (September 10, 2018). Machine learning basics with the k-nearest neighbors algorithm. Towards Data Science. Retrieved January 12, 2025, from (https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761)

WGU College of Information Technology (n.d.). Welcome to D209 Data Mining 1 [PowerPoint slides]. Western Governors University. (https://srm--c.vf.force.com/servlet/fileField?retURL=https%3A%2F%2Fsrm--c.vf.force.com%2Fapex%2FCourseArticle%3Fid%3DkA0S60000001DKzKAM%26groupId%3D%26searchTerm%3D%26courseCode%3DD209%26rtn%3D%252Fapex%252FCommonsExpandedSearch&entityId=ka0S60000006SzJIAU&_CONFIRMATIONTOKEN=VmpFPSxNakF5TlMwd01TMHhNMVF4TnpveU9Ub3hOUzR4TnpkYSx
3D%3D&common.udd.actions.ActionsUtilORIG_URI=%2Fservlet%2FfileField&field=FileUpload__Body__s)