D209 Task 1

Student ID: 012170282

AUTHOR
Tyson Biegler

Part I: Research Question

A1. Can KNN determine which factors are most strongly associated with customer churn, and how accurately can it do so?
A2. The goal is to use KNN to predict customer churn and discover how effective KNN is at predicting customer churn.

Part II: Method Justification

B1: KNN analyzes the dataset by finding the k-nearest neighbors, based on a method like euclidean distance **(GeeksforGeeks, nd)**, and classifying a new data point based on the majority of those nearest neighbors. The expected outcome of this analysis is to identify customers who are likely to churn based on their similarities to other customers.

B2: KNN assumes that data contained in the data set are in closes proximity to other data points that are similar **(Towards Data Science, 2018)**. In other words, similar customers have similar churn outcomes.

B3: The following libraries were used to complete this analysis.

Tidyverse was used for general data wrangling. I also used tidyverse for converting variables to other data types (factor, int, and numeric).

Caret was maybe the most used library. It was used for tasks involved in training the model like `trainControl()` for cross validation, in this case a 10 fold cross validation that was repeated 3 times to evaluate the models performance. Further training was completed with the `train()` function. The train function in caret allowed me to find and select the best K value as well as to use "center" and "scale" to standardize the data. Caret allowed me to evaluate the models performance and make predictions. I was able to generate predictions for the test data with the `predict()` function within caret. Lastly, I was able to evaluate the models performance by creating a confusion matrix using `confusionMatrix()`. This function let me evaluate the sensitivity, specificity, accuracy, and several other measures.

ROCR was used to evaluate and visualize the model's performance by plotting the ROC curve and calculating the AUC. I used `prediction()` to create an object '*perf*' to evaluate the model's performance using the `performance()` function. The performance function allowed me to calculate the true positive and false positive rates, both of which are required for plotting the ROC curve. I used `plot()` to plot the model's performance with a colorized line and a red dashed line, that represents a baseline where the model would be no better than random guessing. After this I calculated the AUC using the performance function again.

Part III: Data Preparation

C1: My data preparation goals were to ensure that the categorical variables were properly encoded as numeric values (0/1) **(R is My Hammer, n.d., *Pre-Processing*)**, and the quantitative variable `Tenure` is scaled appropriately so that KNN can accurately calculate distances. There are also some categorical variables that have multiple unique values that needed to be encoded into dummy variables. I did this using one-hot encoding.

C2: In D208 Task 2 I had created a reduced logistic regression model using Akaike Information Criterion (AIC) and backward elimination. I chose to use those same variables from the reduced logistic regression model because they all had a statistically significant p-value. The selected variables are as follows:

Categorical Variables:

`Churn` (Dependent variable), `Techie`, `Contract`, `InternetService`, `Phone`, `Multiple`, `OnlineBackup`, `DeviceProtection`, `StreamingTV`, `StreamingMovies` and, `PaymentMethod`

Quantitative variable:

`Tenure` (Numeric)

C3: I began by removing all the variables that I was not going to use by selecting only the variables that were included in my reduced model in D208. To ensure the model interprets categorical variables correctly, I converted all binary categorical variables to factors with levels `0` and `1` for "No", and "Yes." Then I created dummy variables for the categorical variables that had multiple categories. Additionally, I ensured that `Tenure` is numeric because a tenure of 30 months should be more similar to 35 months than 10 months. Because of this it is essential that tenure is numeric. Lastly, I standardized the data because KNN is based on distance calculations and a large tenure could dominate the distance calculation. In my code below the standardization (z-score standardization), in which each of the variables have a mean of 0 and a standard deviation of 1 **(Buya, 2023)**, is accomplished with the line `churn$Tenure <- as.numeric(scale(churn$Tenure))`. `Scale()` will return a matrix by default so I ensured that it is converted to numeric after scaling.

C4: The cleaned data set will be included in my submission files and will be named `CLEANED_churn.csv`.

Part IV: Analysis

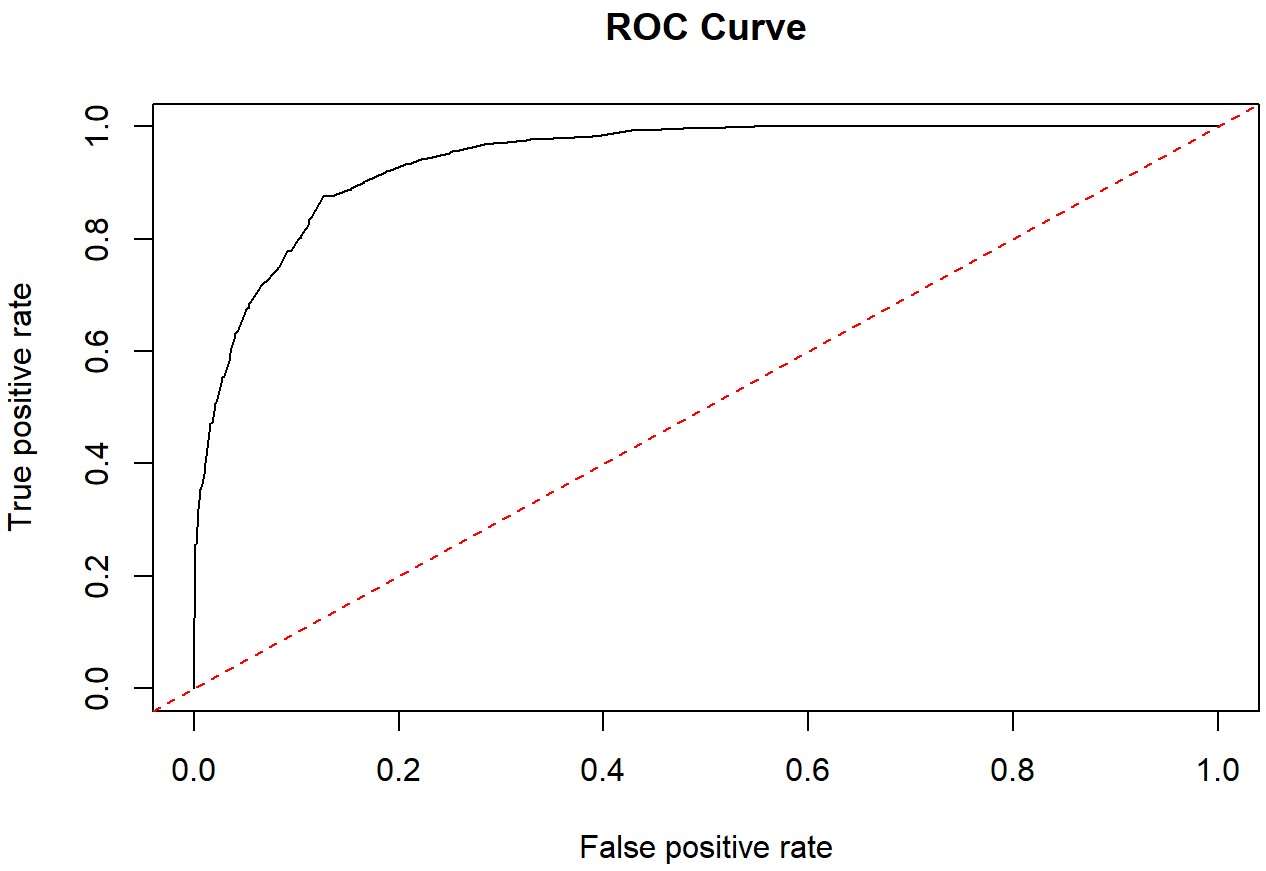
D1: I split the data into training and test data with an 80/20 split.

D2: I used KNN to predict customer churn based on factors like `Techie`, `Contract`, `InternetService`, `Phone`, `Multiple`, `OnlineBackup`, `DeviceProtection`, `StreamingTV`, `StreamingMovies` and, `PaymentMethod`. To improve accuracy I applied cross validation with the `method="repeatedcv"`. This 10 fold cross validation process is repeated 3 times.

Before training, I standardized the quantitative variable using `churn$Tenure <- as.numeric(scale(churn$Tenure))`. This ensures that this particular values does not dominate the distance

measurements the KNN relies on because the other variables are all binary at this point. I used the `predict()` function to make predictions using new data stored in the training set.

To further evaluate the model I created a ROC curve, which shows how well the model distinguishes between customers who churn and those who do not. Along with the ROC, I also calculated the AUC to confirm the model's performance.



D3: The code used explained in section D2 is provided below.

Part V: Data Summary and Implications

E1: AUC ranges from 0.5 - 1. Values close to 1 indicates that the model is accurately separating churned and non-churned customers. In this case, the AUC score is 0.9422046 meaning that the model has a strong ability to rank customers based on their likelihood of churning.

E2.

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  1382  140
       Yes   99  365
```

```
               Accuracy : 0.8797
                 95% CI : (0.8645, 0.8937)
    No Information Rate : 0.7457
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.674

 Mcnemar's Test P-Value : 0.009671

            Sensitivity : 0.9332
            Specificity : 0.7228
         Pos Pred Value : 0.9080
         Neg Pred Value : 0.7866
             Prevalence : 0.7457
         Detection Rate : 0.6959
   Detection Prevalence : 0.7664
      Balanced Accuracy : 0.8280

       'Positive' Class : No
```

```
[1] "AUC : 0.943"
```

This model has an accuracy of 87.97%, meaning that it can correctly predict the churn status of a custom the majority of the time. This models accuracy is better than the No Information Rate of 74.57%, meaning that it performs significantly better than if the model just predicted based on the majority class. The p-value of < 2.2e-16 confirms that this model's predictions are not due to random chance. The sensitivity rate of this model is 93.32%, meaning that the model is very good at identifying the true negatives, or the customers who did not churn. In contrast, 72.28% (specificity) of the time, the model correctly identified the true positives, the customers who did churn.

Despite the relatively low specificity, the model has balanced predictive values, meaning that in real world predictions this model does a good job correctly predicting the churn status of customers. When the model predicts a customer will not churn, it is correct 90.80% of the time (`Pos Pred Value`). Similarly, when the model predicts that a customer will churn, it is correct 78.66% of the time (`Neg Pred Value`).

Considering the accuracy rate of 87.97% and the AUC score of 0.9422046, I can conclude that in the majority of cases, this model will correctly predict or identify the churn status of a customer. Businesses can use this model to proactively target customers who are at risk for churn.

E3. One limitation of this analysis is the use of mostly binary variables in the model. Since KNN works with distance measurements (euclidean distance in this case), it works best with continuous variables where the variables have a wider range of distance. But with Binary variables with only 1 or 0 as their values, the distances are shorter and lose precision. In this KNN model I have only one continuous variable, `Tenure`, and the rest are binary.

E4. Based on the model's performance, the organization should focus on proactive retention strategies for at-risk customers. With 87.97% and the AUC score of 0.9422046, the model effectively predicts churn, allowing the company to intervene early. Since it correctly identifies customers who churn 72.28% of the

time, the company should use these insights to offer personalized discounts, improved support, or loyalty incentives to address the customers who are at risk of churning.

Part VI: Demonstration

F. My panopto video link will be provided in the submission files.

Sources

1. **Buya, A. (2023, July)**. *The fundamentals of k-nearest neighbors: Normalization and standardization*. Medium. Retrieved from https://medium.com/@buyaalfariz/the-fundamentals-of-k-nearest-neighbors-normalization-and-standardization-a3e6ca616d57
2. **Towards Data Science. (2018, September)**. *Machine learning basics with the K-nearest neighbors algorithm*. Retrieved from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
3. **GeeksforGeeks. (n.d.)**. *K-nearest neighbours*. Retrieved from https://www.geeksforgeeks.org/k-nearest-neighbours/#
4. **R is My Hammer. (n.d.)**. *Pre-processing in machine learning*. Retrieved from http://rismyhammer.com/ml/Pre-Processing.html