STAT 642 - Final Project

Case #4: Predicting Customer Churn

Group 8

Group Members: Prateek Gulati, Sadhana Manekar, Frank McDermott, Harshit Sanwal

March 15, 2021

**Executive Summary**

Customer churn has a huge impact on the telecom industry. The long-term profitability of a company depends on the relationship of the organization with its customers. Companies can potentially increase revenue if they can predict customers who are likely to churn and take necessary actions to reduce it. By reducing churn, our company can save costs on new customers, build long-term customer relations and develop insights on customer retention.

In this project we predicted whether a customer would churn using unsupervised learning methods, such as kmeans and hierarchical cluster analysis, and supervised classification, such as naïve bayes, ensemble methods, and k-nearest neighbors. Our analysis ultimately led us to create 3 models, each with its own set of variables that it designated as the most important in predicting churn. These variables are as follows.

- Contract
- OnlineSecurity
- TechSupport
- OnlineBackup
- InternetService
- Dependents
- Tenure
- MonthlyCharges

We have found that the ideal way to predict whether a customer will churn is by using a mixture of these variables. By analyzing these individual variables, it is clear that each variable has a potential value that leads to a much higher likelihood of churn. The input for each variable that has the highest potential for customer churn is listed below.

- Contract: Month-to-month
- OnlineSecurity: No Online Security
- TechSupport: No Tech Support
- OnlineBackup: No Online Backup
- InternetService: Fiber Optic
- Dependents: No Dependents
- Tenure: 0 to 20 years (shorter tenures)
- MonthlyCharges: $50 to $80 per month (higher monthly charges charges)

We suggest building a list of customers that meet most of this criteria and performing outreach. If our workforce is capable of it, we can begin by having our representatives speak to these customers that are at high-risk to churn and determine ways to boost their satisfaction. If this will be too demanding in terms of cost and time, an easier, more standardized solution would be to send out online surveys to these customers and offer a chance at a prize as an incentive.

Our goal is to reduce churn as much as possible. Now that we have identified our high-impact variables, we can begin to target our high-risk customers and do whatever we can to stop them from leaving our service.

## Introduction

Customer churn is a major problem for telecom companies. Customers leave their telecommunications company frequently because they are dissatisfied with network strength, tariffs, attractive offers from competitors etc., This phenomenon of loss of customers is referred to as Customer Churn or Customer Attrition. Even with low percentages of customer churn, companies can lose millions of dollars every month.

Deploying unsupervised and supervised machine learning will help build a prediction model and classification model to ease out the process of customer retention for the company. We used the data on past and present customers to help us better understand them, predict whether they will churn or not, accordingly, grouped/classified them and identified the factors causing the churn in order to curb the attrition and in turn to avoid the potential revenue loss. It is costly to the company, losing a good customer. Our objective is not only to predict but to minimize cases where we predict a customer will stay and they actually leave. These prediction and classification models will help the company in taking proactive measures to retain the customer by tailoring the plans as per customer needs', thereby maintaining a significant revenue stream.

## Dataset

### 1. Data Description

Our analysis methods will include supervised and unsupervised methods on the given volume of the data set which include 7043 observations with 21 variables. Data provided is of high quality with no duplicate values and only 11 missing values "Total Charges" variable.
Snapshot of variables listed in the dataset:

Table 1: Our Variables

| Variable Type | Variable Name | Additional Notes |
|---|---|---|
| Nominal Variable | Churn | Dependent Variable |
| | Contract | MonthToMonth, OneYear, TwoYear |
| | CustomerID | Not a useful predictor |
| | Dependents | Yes/No |
| | Device Protection | Yes, No, NoInternetService |
| | Gender | An approximately equal mix of Male and Female |
| | InternetServices | DSL, FiberOptic, No |
| | MultipleLines | Yes, No, NoPhoneService |
| | OnlineBackup | Yes, No, NoPhoneService |
| | OnlineSecurity | Yes, No, NoPhoneService |
| | PaperlessBilling | Yes/No |
| | Partner | Yes/No |
| | Payment Method | BankTransfer, CreditCard, ElectronicCheck, MailedCheck |
| | PhoneService | Yes/No |
| | SeniorCitizen | Represented as Binary |
| | StreamingMovies | Yes, No, NoPhoneService |
| | StreamingTV | Yes, No, NoPhoneService |
| | TechSupport | Yes, No, NoPhoneService |
| Numeric variable | Monthly Charges | [$18.25, $118.75] |
| | Tenure | Represented as Integers |
| | Total Charges | [$18.80, $8684.80] with 11 missing values |

### 2. Data Cleaning and Preprocessing

The algorithms developed have a prerequisite that the dataset should not have missing values and redundant variables.
Missing values

Removed from the dataset which were present primarily in the "Total Charges" variable.

Redundant Variables

1. Total Charges: Correlation among numerical variables (Total Charges, Monthly Charges, Tenure) was checked, and removed the highly correlated variable i.e."Total Charges".
2. Removed CustomerID variable as it is not a useful predictor of customer churn.

Final dataset contained 7032 observations with 19 variables.

Data has been standardized the data using center and scale, Yeo-Johnson Transformations were used to normalize the data. Further, the dataset was partitioned into a training and testing split with the ratio of 85:15. Distribution of variable predicting i.e. Customer Churn has been preserved in the training and testing datasets.

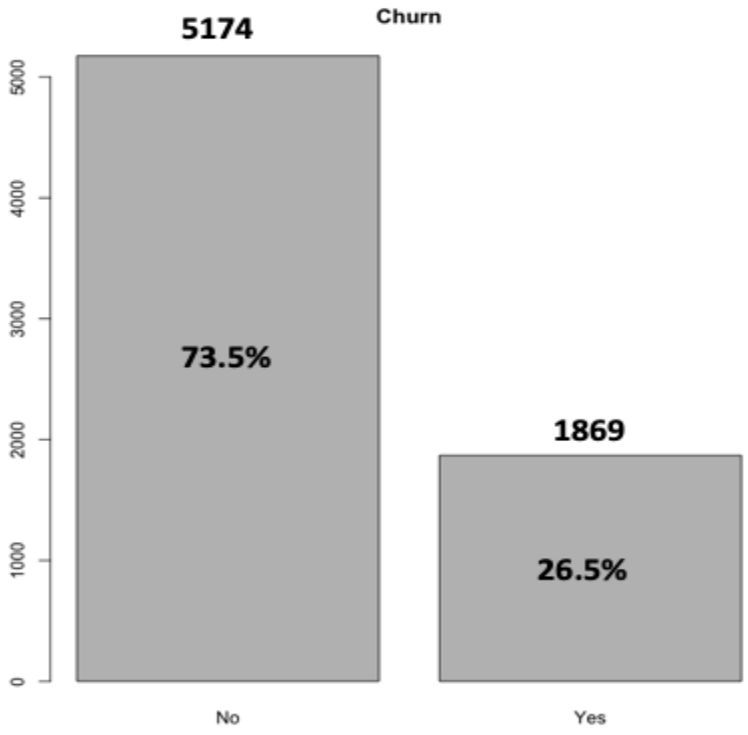
Seed value used 29619122 throughout our analysis, in order to reproduce the results.

## 3. Descriptive Statistics

Before deep diving into the analysis, below mentioned are the general insights from the data about the variable in focus "Customer Churn".

Table 2 below showcases 26.5 % of total customers have left the company and are represented by "Yes" and those who stayed (73.5%) are represented by "No". Yes and No are labelled in terms of whether they churned or not.

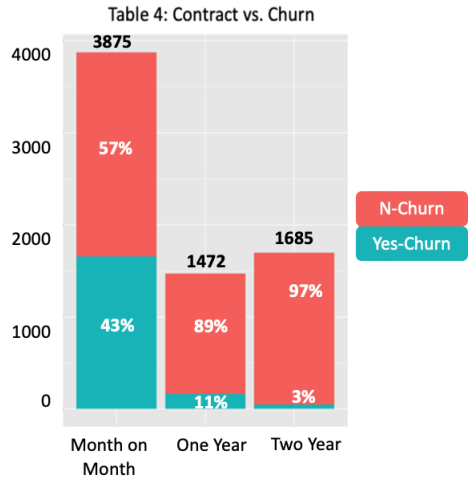
Table 2: Customer Churn (Yes vs. No)

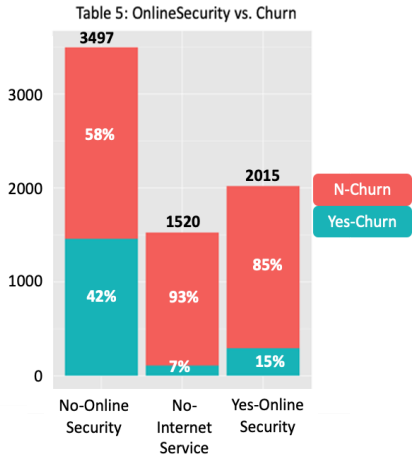


The effect of individual variables on Churn is showcased below.

Categorical Variables

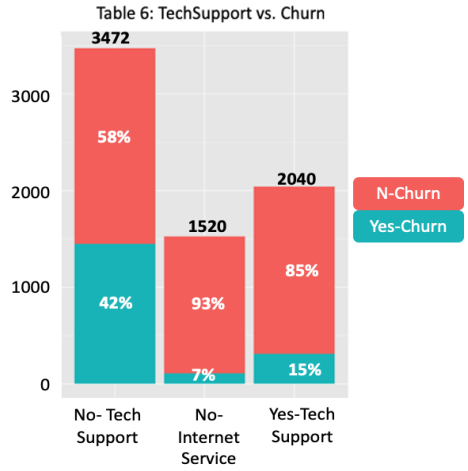
There is no gender biasness w.r.t churn as males and females have equal levels of churn.

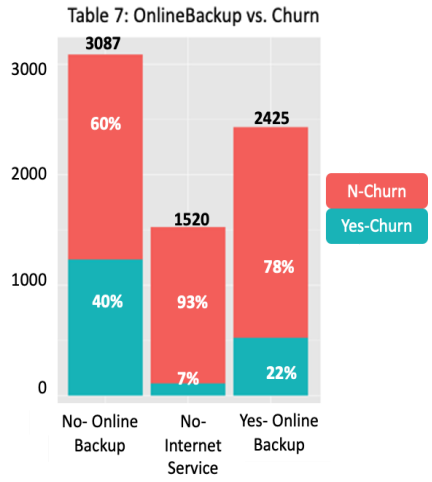



As the length of the contract increases, the number of churned customers decreases, meaning that the relation between contract and churn is inversely proportional.

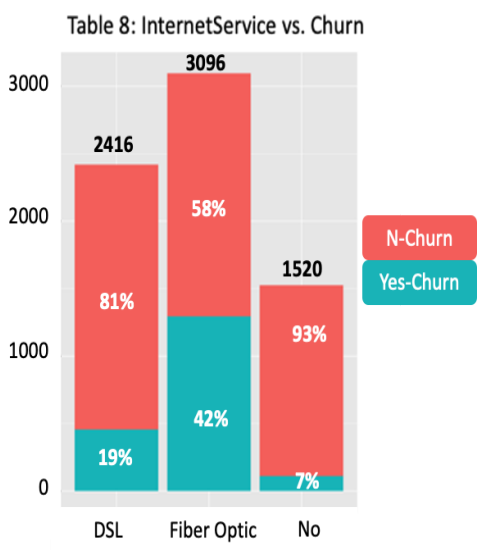
Customers giving more importance to online security are likelier to churn.

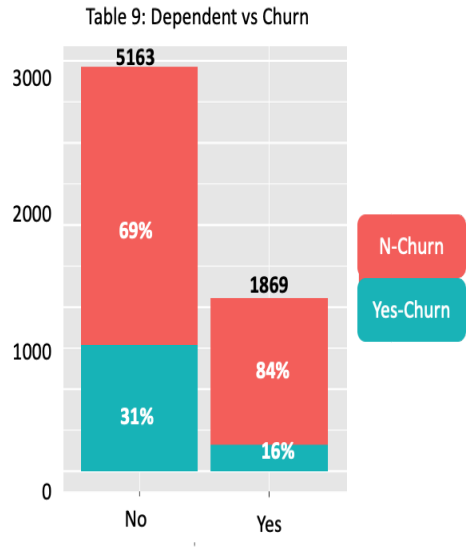



Customers giving more importance to tech-support are likelier to churn.

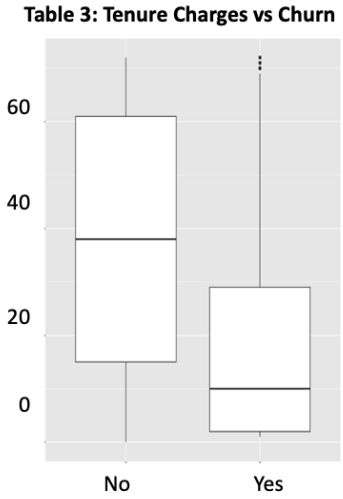
Customers giving more importance to online backup are likelier to churn.

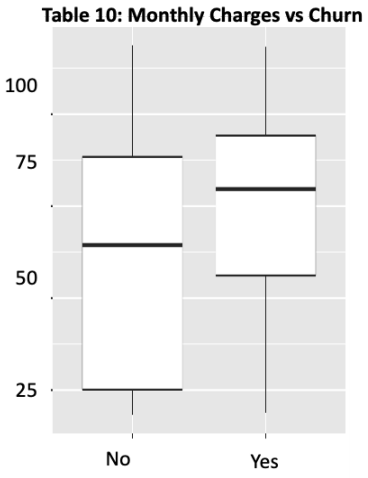



Customers having DSL infrastructure as an internet service are likelier to churn. One third of the customers having dependents at home are likelier to churn.

Numerical Variables




Customers who left the company tend to have shorter tenures.
Customers with higher monthly charges are likelier to leave the company than those with higher total charges.

## Analysis

We will use kMeans and Hierarchical Cluster Analysis for classification of customers, K-Nearest Neighbour to predict the customer churn and lastly, to identify the reasons for the churn, we will use Naïve Bayes and Random Forest models.

With the help of our data and EDA patterns we would like to model customer churn, using two methods:

1. Unsupervised learning methods
2. Supervised/Classification methods

### Unsupervised Learning Methods

In our analysis, we will use kMeans and Hierarchical clustering analysis to categorize the customers that behave similarly. This way, our telecom business can understand the behavior of the customers, segment the markets, and analyze them to optimize their processes within each cluster/group.

As the given dataset is a mixture of categorical and numerical variables, we have used the 'Gower' method of computing the distances between each pair of observations and defining the (dis)similarities among the clusters.

In the KMeans clustering analysis, we have initially assigned 5 as the number of clusters. The results shown below (table 11) are the cluster assignments for each observation. Out of all the clusters, cluster 3 has the greatest number of customers with 1807.

From the visualization of KMeans (table 12) cluster solution, we can see that the 5 clusters are plotted based on MonthlyCharges and tenure.

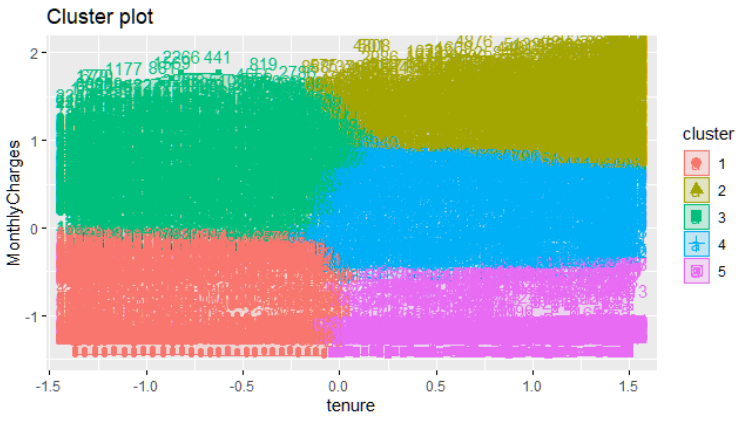
Below are the inferences from the visualization:

- Customers with low tenure and high monthly charges belong to Cluster 3 (Green)
- Customers with low tenure and low monthly charges belong to Cluster 1 (Orange)
- Customers with high tenure and high monthly charges belong to Cluster 2 (Pale Green)

Table 11: Clustering Vector

```
Clustering vector:
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
   3   1   3   4   5   5   5   3   2   1   3   3   2   2   5   2
  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
   4   2   3   5   3   3   3   1   1   1   2   3   2   5   2   5
  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48
   1   3   3   2   5   1   2   5   3   1   3   1   5   1   3   5
  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64
   1   1   2   5   5   5   1   5   2   2   4   2   2   2   4   3
  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
   5   5   1   1   4   5   3   4   2   4   5   2   1   3   1   4
  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96
   5   3   5   2   4   5   1   4   4   3   5   5   2   2   2   5
  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112
   1   3   4   5   3   3   2   4   2   3   2   4   2   4   2   1
 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128
   2   1   1   5   4   5   4   5   2   3   5   3   3   1   5   4
 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
   3   1   5   3   1   3   3   4   5   4   4   5   2   5   2   2
 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
   4   2   3   3   1   5   5   5   2   2   3   5   5   3   5   3
 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176
   3   5   5   1   3   4   5   2   4   3   1   5   5   2   4   4
 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192
   5   5   5   4   3   3   5   1   5   3   3   3   3   1   1   5
 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208
   4   2   4   5   3   2   2   3   5   5   2   5   1   1   1   5
```

Table 12: Visualization of kMeans




From the below table (table 13), which shows the distribution of customer churn in all 5 clusters, it is clear that the number of customers who churned is higher than the number of customers who did not churn in cluster 3.

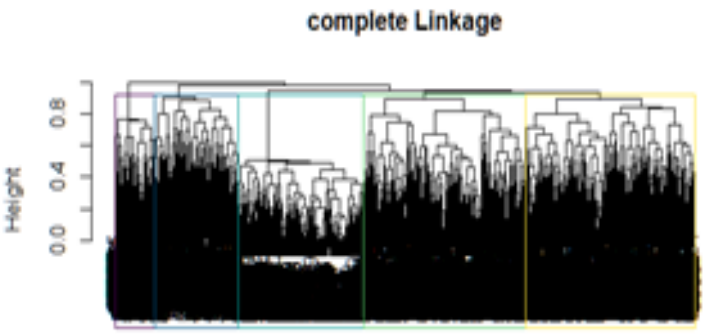
Table 13: Churn vs Clusters

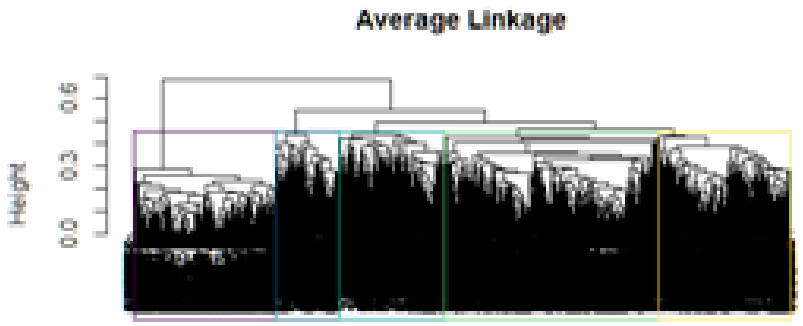
| | Clusters | | | | |
|---|---|---|---|---|---|
| **Customer Churn** | **1** | **2** | **3** | **4** | **5** |
| **No** | 1291 | 1181 | 829 | 998 | 864 |
| **Yes** | 435 | 276 | 978 | 150 | 30 |

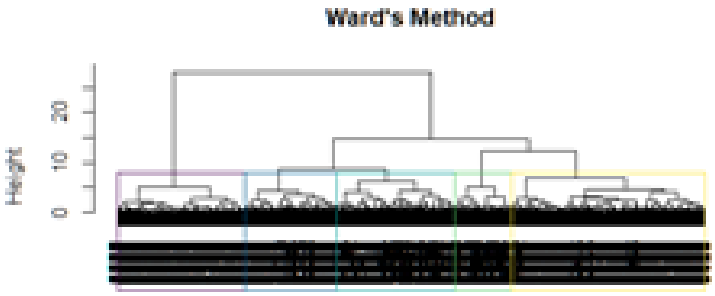
From Table 12 and Table 13 we can conclude that the company is experiencing huge customer churn as monthly charges are high for customers with low tenure.

The potential disadvantage of KMeans is that it requires us to pre-specify the number of clusters, where in HCA it is not necessary to commit to a certain number of clusters. Additionally, HCA results in a 'Dendrogram', an attractive tree-based representation of the observations.

In HCA we focused on the agglomerative type of clustering to calculate the proximities between clusters using Single, Complete, Average, Centroid and Wards linkage methods. From the resultant dendrograms, we could classify the number of clusters as 5.




The average linkage method has the highest Cophenetic Correlation (0.79) when compared to other linkage methods, indicating stronger clustering structures.

To compare KMeans and HCA, we have used several validation measures such as Within-Cluster sum of squares, Average Silhouette width, Intra-cluster distance, Inter-cluster distance, Dunn Index, and Average distance within and between the clusters.

The optimal number of clusters for HCA using the within-cluster sum of squares method is 5, which is the same as our initial classification of clusters from the dendrogram. While in terms of average silhouette, the number of clusters is only 2 (see table 14 and table 15).

Table 14: WSS Method (HCA)

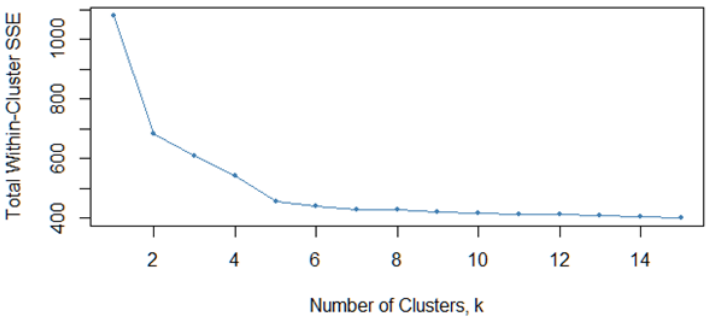
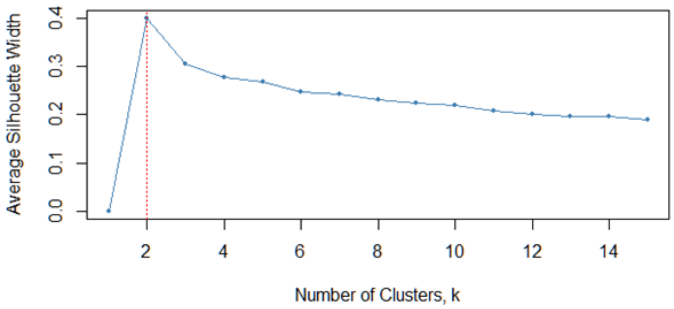



Table 15: Silhouette Method (HCA)




By contrast, according to KMeans cluster analysis, the optimal number of clusters varied from our initial assumption. The intra-cluster variation is minimum at 4 and the average silhouette width is maximum with 4 clusters (see table 16 and table 17).

Table 16: WSS Method (kMeans)

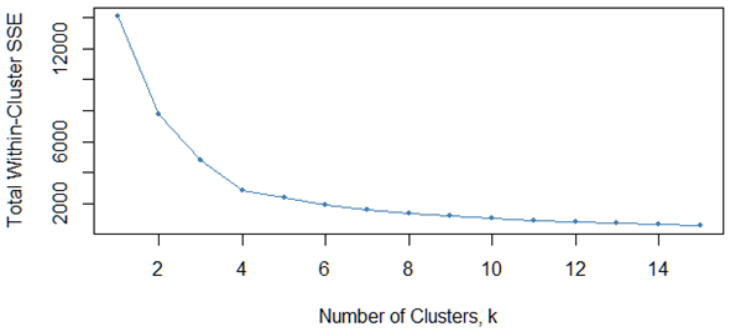
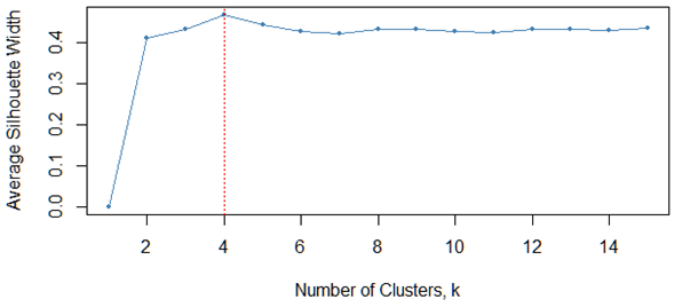



Table 17: Silhouette Method (kMeans)



Below (table 18) are the results of the other validation measures and it is clear that HCA has done a better job in clustering in many measures, such as separation between the clusters, diameters of the clusters, average distance between the clusters and dunn index (minimum separation/maximum diameter).

Table 18: Other Validation Measures

| Validation Measures | HCA | KMeans |
|---|---|---|
| **max.diameter** | 0.8831526 | 2.304066 |
| **min.separation** | 0.05573882 | 0.008606 |
| **average.between** | 0.5828156 | 2.13194 |
| **average.within** | 0.3366708 | 0.809427 |
| **dunn (Dunn Index)** | 0.06311346 | 0.003735 |

**Supervised Learning Methods**

Naïve Bayes

Naïve Bayes Classification is a technique that assumes independence between predictor variables. In other words, Naïve Bayes Classification works best when independent variables cannot be used to accurately predict other independent variables. We have chosen to use this method of analysis as it will individually exam each independent variable and determine if it has any bearing on customer churn.

When performing our analysis, we found that the variable TotalCharges (the total amount that the customer has been charged over their lifetime) strongly correlates with our other two numeric variables, Tenure (their total time as a customer) and MonthlyCharges (the most recent amount that they have been charged). To better suit the Naïve Bayes Classification method, TotalCharges was removed from our analysis. We also normalized our data for MonthlyCharges and Tenure for our analysis.

Next, we divide our data into training and testing datasets. Our training dataset takes 85% of our observations and builds our model, while the testing dataset measures the ability of that model to predict the remaining 15% of our observations. We will use our two datasets to generate performance measures that we can compare to other models to determine our ideal analysis method.

First, we will generate our overall model.

Table 19: Naïve Bayes Performance Measures: Overall

| | Training | Testing |
|---|---|---|
| Accuracy | 0.7278 | 7.45E-01 |
| Kappa | 0.4228 | 4.46E-01 |

The values of interest in this table are Accuracy and Kappa. Accuracy measures the proportion of correct predictions, meaning that this model correctly predicts customer churn 74.5% of the time, which appears to be decent. Our Kappa Statistic measures accuracy while also accounting for the possibility of the correct predictions being obtained merely by chance. There are only two possible outcomes for churn, yes or no, so if our model were to make absolutely random predictions, it would be correct 50% of the time. Our Kappa Statistic of 44.6% indicates a moderate agreement between our model and the actual results.

Next, we will generate our class-level model.

Table 20: Naïve Bayes Performance Measures: Class-Level

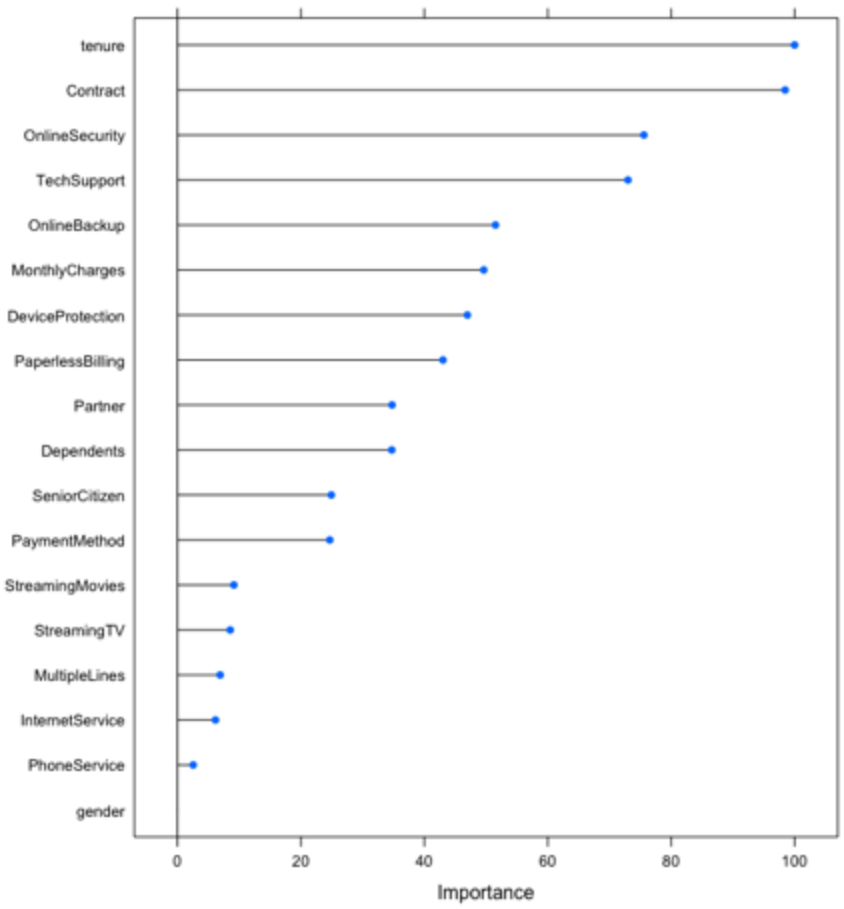
| | Training | Testing |
|---|---|---|
| Sensitivity | 0.8143486 | 0.8107143 |
| Specificity | 0.6926407 | 0.7093023 |
| Pos Pred Value | 0.4927647 | 0.5022124 |
| Neg Pred Value | 0.9119928 | 0.9119601 |
| Precision | 0.4927647 | 0.5022124 |
| Recall | 0.8143486 | 0.8107143 |
| F1 | 0.6139976 | 0.6202186 |
| Prevalence | 0.2658080 | 0.2656546 |
| Detection Rate | 0.2164604 | 0.2153700 |
| Detection Prevalence | 0.4392774 | 0.4288425 |
| Balanced Accuracy | 0.7554313 | 0.7600083 |

Our variables of interest in this model are Sensitivity, Specificity, and F(1)-Measure. Sensitivity measures the proportion of true positives. When the observed customer leaves our service, our model correctly predicts it 80.4% of the time. Specificity measures the proportion of true negatives. When the observed customer stays with our service, our model is able to correctly predict it 72.4% of the time. Our F(1)-Measure is a goodness of fit assessment that is calculated based on both Precision (the proportion of true positives over total predicted positives) and Recall (the proportion of true positives over the total amount of positives) in equal measure. The F(1)-Measure indicates that Precision and Recall balance out to a proportion of 62.6%.

In both our overall and class-level analyses, our testing and training datasets behaved similarly in all performance measures. This means that we have generated a balanced model.

To finish analyzing our Naïve Bayes model, we have generated a table that determines the importance of each variable in descending order. Based on our findings, we believe that Tenure, Contract, OnlineSecurity, TechSupport, and OnlineBackup are the most important variables to consider when determining whether a customer will leave our company.

Table 21: Naïve Bayes Variable Importance




Random Forests

Random Forest analysis is an ensemble method that trains several decision trees in parallel with bootstrapping followed by aggregation. Bootstrapping indicates that several individual decision trees are trained in parallel on various subsets of the training dataset using different subsets of available features. Bootstrapping ensures that each individual decision tree in the random forest is unique. The random forest classifier aggregates the decisions of individual trees to produce a more accurate prediction.

To build the random forest model, we cleaned missing values from our data and removed the unnecessary variable customerID.

We further partition our data into training and testing datasets. Like the Naïve Bayes model, our training dataset takes 85% of our observations and builds our model, while the testing dataset measures the ability of that model to predict the remaining 15% of our observations.

After partitioning our data we start building the random forest model for and run it on the training data set.

After getting the output from our base random forest model, we will try to improve it by tuning our model.

Now by comparing the values of the confusion matrix that we generated for both models, we get the following table.

Table 22: Random Forest Training vs. Testing

| | Training | Testing |
|---|---|---|
| Accuracy | 0.7911 | 0.8055 |
| Kappa | 0.432 | 0.4563 |
| Sensitivity | 0.9031 | 0.9147 |
| Specificity | 0.4964 | 0.5036 |
| OOB Error Rate | 21.34% | 19.96% |

Our variables of interest in this model are Accuracy, Kappa, Sensitivity, Specificity and OOB error rate. Accuracy measures the proportion of correct predictions, meaning that this model correctly predicts customer churn 80.5% of the time, which is an improvement on Naïve Bayes. Our Kappa Statistic measures accuracy while also accounting for the possibility of the correct predictions being obtained merely by chance. Our Kappa Statistic of 45.6% indicates a moderate agreement between our model and the actual results, another improvement on the Naïve Bayes model.

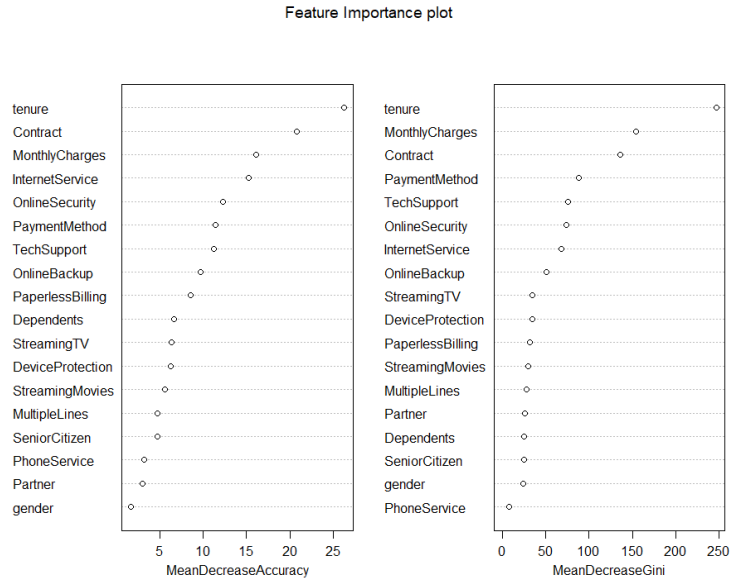
Sensitivity measures the proportion of true positives. When the observed customer leaves our service, our model correctly predicts it 91.4% of the time. Specificity measures the

proportion of true negatives. When the observed customer stays with our service, our model is able to correctly predict it 50.3% of the time. Our sensitivity improves on the Naïve Bayes model, while specificity degrades.

Out-of-bag (OOB) error is a method of measuring the prediction error of random forests. We grow the tree on a bootstrap sample ("the bag"). About two-thirds of the cases are in the bag. The remaining one-third are "out-of-bag". The out-of-bag data is like a test set for this tree – by passing them down the tree, we can compute their error rate. Our tuned model has a OOB error of 19.96% which has decreased from our base model.

Now we will plot the feature importance plot to derive the top features having an impact in customer churn.

Table 23: Random Forest Variable Importance




Based on our findings from the feature importance plot , we believe that Tenure, Contract, Monthly Charges and Internet Service are the most important features to consider when predicting customer churn. Some features, such as gender, have little to no impact on customer churn.

KNN

Accuracy for the KNN model was 80% and Kappa was 44%. 9 out of 10 customers who were predicted to stay by the model ended up staying (sensitivity), while 5 out of 10 customers predicted to churn by the model ended up churning (specificity).

Table 24: KNN Performance Measures

| | Basic | Actual | |
|---|---|---|---|
| | # Customers | Stay | Churn |
| Predicted | Stay | 68% | 14% |
| | Churn | 5% | 12% |

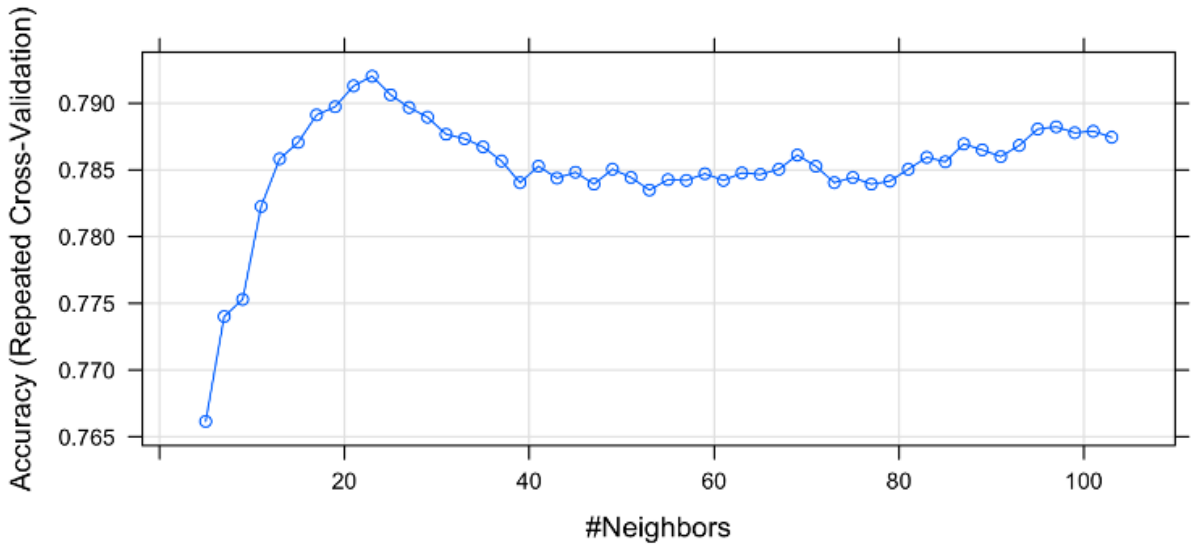
Model: When a new customer is introduced to the model, it will reference the database, identify the most similar customers, and then predict if the new customer would churn based on whether similar customers churned.

After removing the missing values, the dataset has 7032 unique customers, which were further split into training and testing dataset in the ratio of 85:15. 1869 of these customers churned while 5163 stayed with the company.

**Model Performance:**

Repeated cross validation confirms that accuracy is best around K=21

Table 25: KNN Accuracy




Confusion Matrix: Highlighted parameters suggest that the base model has high accuracy, but the Kappa suggests it's a moderate model as it takes into account the possibility of correct prediction by chance.
Goodness of fit: Overall, it is a balanced model.

Table 26: KNN Confusion Matrix

| | KNN | |
|---|---|---|
| Overall | Base | Tuned |
| Accuracy | 0.8045541 | 0.7941176 |
| Kappa | 0.4375259 | 0.4186505 |
| Class | Base | Tuned |
| Sensitivity | 0.4607143 | 0.4678571 |
| Specificity | 0.9289406 | 0.9121447 |
| Pos Pred Value | 0.701087 | 0.6582915 |
| Neg Pred Value | 0.8264368 | 0.825731 |
| Precision | 0.701087 | 0.6582915 |
| Recall | 0.4607143 | 0.4678571 |
| F1 | 0.5560345 | 0.5469729 |

Model Comparison

Table 27: Model Comparison

| | KNN | | Naïve Bayes | | Random Forest | |
|---|---|---|---|---|---|---|
| Overall | Base | Tuned | Training | Test | Training | Test |
| Accuracy | 8.05E-01 | 7.94E-01 | 7.28E-01 | 7.36E-01 | 7.91E-01 | 0.8055 |
| Kappa | 4.38E-01 | 4.19E-01 | 4.23E-01 | 4.35E-01 | 4.32E-01 | 4.56E-01 |
| Class | Base | Tuned | Training | Testing | Training | Testing |
| Sensitivity | 0.4607143 | 0.4678571 | 0.8143 | 0.8107 | 0.9031 | 0.9147 |
| Specificity | 0.9289406 | 0.9121447 | 0.6965 | 0.7093 | 0.4964 | 0.5036 |
| Pos Pred Value | 0.701087 | 0.6582915 | 0.4928 | 0.5022 | 0.8289 | 0.8295 |
| Neg Pred Value | 0.8264368 | 0.825731 | 0.912 | 0.912 | 0.6412 | 0.6504 |
| Precision | 0.701087 | 0.6582915 | 0.4928 | 0.5022 | | |
| Recall | 0.4607143 | 0.4678571 | 0.8143 | 0.8107 | | |
| F1 | 0.5560345 | 0.5469729 | 0.614 | 0.6202 | | |

**Conclusion**

All three of the models that we built have similar accuracy and kappa-values. We can only recommend a model once we know what our most important performance measure is. If we are looking for the best goodness of fit or recall, we would suggest Naïve Bayes. If we want to maximize sensitivity, we would suggest the Random Forest Method. And if we want to maximize specificity or precision, we would recommend k-Nearest Neighbors. All of the supervised methods offer their pros and cons, so we cannot definitively recommend one over another. We can, however, suggest that we focus on the results of our supervised models over our unsupervised ones.

Based on this, we instead suggest focusing on the most important variables designated by each of the models, as each model has some validity to it. These variables are detailed in the

descriptive statistics section and are represented in tables 3 through 10. The rankings of these important variables are detailed in tables 21 and 23.

For example, tenure was listed as an important variable in multiple models. Generally speaking, customers who have been with us for a shorter period of time are likelier to churn. Our models suggest that our priority should be to reach out to our shorter tenured customers and find ways to increase their satisfaction with our service. The same applies to customers who have month-to-month contracts and no online security, online backup, or tech support. By reaching out to these customers, and figuring out ways to increase their satisfaction, we will undoubtedly lower our churn.

References


1. "Determine Variables of Importance in Naive Bayes Model". *Stack Overflow*. July, 2020. https://stackoverflow.com/questions/62849037/determine-variables-of-importance-in-naive-bayes-model.
2. Joshi, Pragya. Gupta, Surenda. "Predicting Customers Churn in Telecom Industry using Centroid Oversampling method and KNN classifier." International Research Journal of Engineering and Technology (IRJET). Vol. 6, Issue 4. April, 2019. https://www.irjet.net/archives/V6/i4/IRJET-V6I41166.pdf.
3. Brandusoiu, Ionut. Toderean, G. "Predicting Churn in Mobile Telecommunications Industry." *ResearchGate*. July, 2013. https://www.researchgate.net/publication/334051845_Predicting_Churn_in_Mobile_Telecommunications_Industry.
4. Jawaharlal, Vijayakumar. "kNN Using Caret R Package." *RStudio-Pubs*. April 29, 2014. https://rstudio-pubs-static.s3.amazonaws.com/16444_caf85a306d564eb490eebdbaf0072df2.html.
5. Zach. "How to Build Random Forests in R (Step-by-Step)." *Statology*. November 24, 2020. https://www.statology.org/random-forest-in-r/#:~:text=How%20to%20Build%20Random%20Forests%20in%20R%20(Step-by-Step),contains...%20Step%203:%20Tune%20the%20Model.%20By.