Iustin Toader, Ryan Keon

Drexel University

STAT-331-130

March 14, 2021

Introduction

The loss of valuable customers hinders TK Telecom's (TKT) ability to maximize customer retention, thus negatively affecting revenue. Even a low customer churn rate can substantially diminish projected revenue growth. This report delineates our analysis of the customer churn rate of TKT using the given data of both past and present customers. Through predictive analysis, we aim to predict future customer churn based on the given variables. With this information, TKT will be better prepared to take appropriate measures to increase customer retention.

Data

During our initial examination of TKT's customer database, we noted 2114 observations of 14 variables. In order to organized these given variables we then categorized them based on their variable types accordingly:

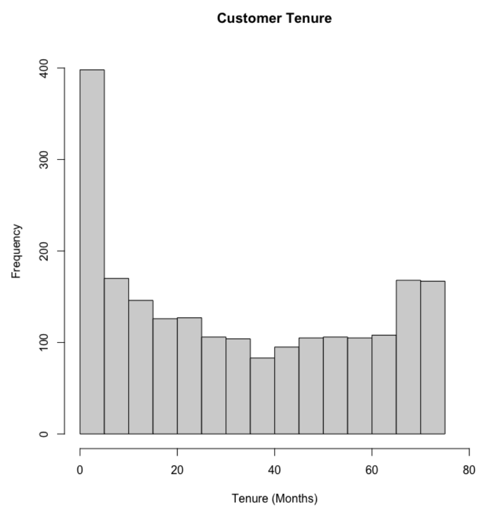
- Nominal: *customerID, gender, SeniorCitizen, Partner, Dependents, PhoneService, InternetService, PaperlessBilling, PaymentMethod, Churn*
- Ordinal: *Contract*
- Numerical: *tenure, MonthlyCharges, TotalCharges*

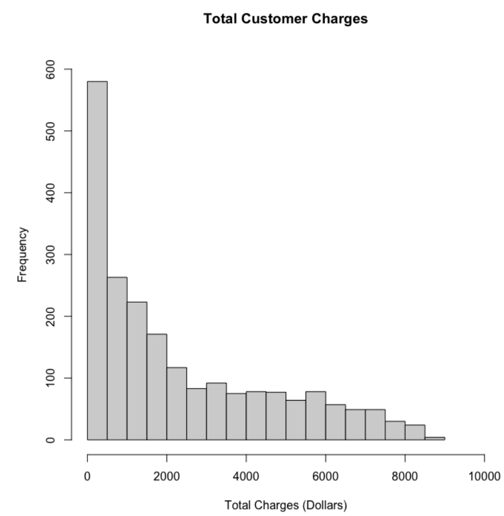
To prepare the data for analysis, we identified and appropriately addressed any missing values, duplicate values, outliers, and redundant variables. While the quality of the data within the database is robust we identified two missing values within the total charges variable, however, this only occurred for new customers with a tenure of zero. As a result, we imputed these missing values with the median value of all other total charges. We concluded that there are no duplicate values as well as no considerable outliers that would interfere with our analysis moving forward. Finally, we chose to remove the customer ID variable as it has no relevance to our analysis. With this now cleaned and transformed data, we determined the churn variable as our target variable as our main objective is to predict customer churn based on the other independent variables.

(Fig.1)

(Fig. 2)

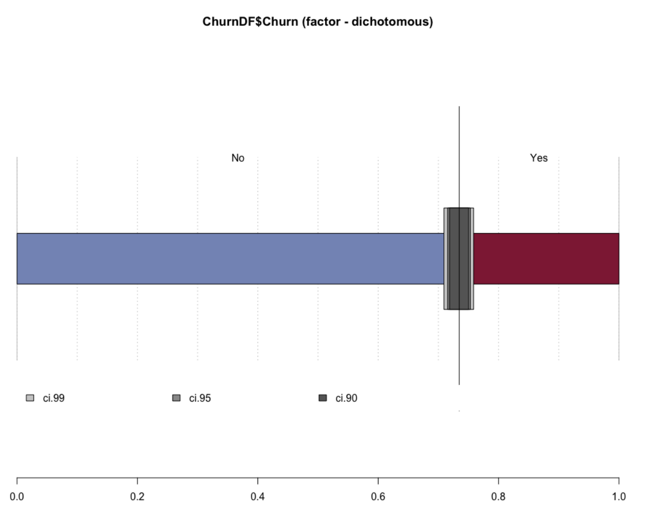
```
13 - TotalCharges (numeric)

  length        n      NAs    unique       0s       mean     meanCI'
   2'114    2'112        2     2'047        0  2'337.342  2'239.896
```






```
heap(?): remarkable frequency (8.9%) for the mode(s) (= 1)
```

(Fig. 3)



(Fig. 4)

Analysis

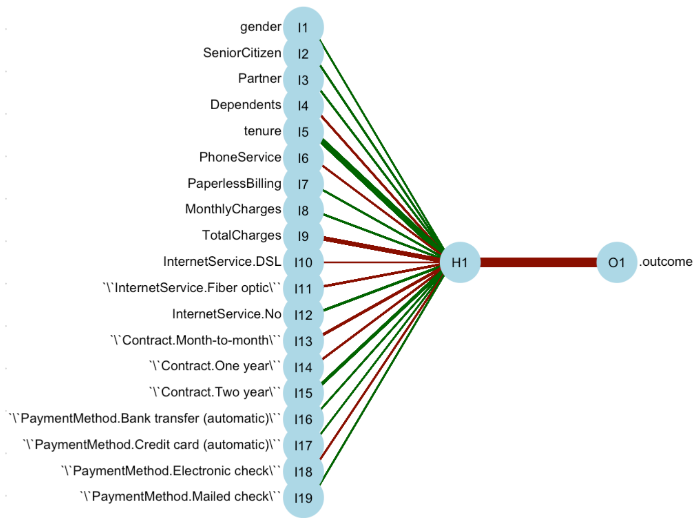
We should first mention that we decided to use two supervised learning methods because our dataset contains 3 numerical variables out of the 14 total variables. Since the unsupervised methods of Clustering and Principal Components Analysis only work with numerical variables, we decided that the analysis would not have been representative of our dataset, nor would have generated any actionable insights. We also decided that Association Analysis would not have been an appropriate method for our dataset.

Artificial Neural Networks

The first analysis which was conducted was a supervised classification method. Specifically, we used an Artificial Neural Network to predict the customers which would leave the company. We chose this method because of the apparent complexity between the well-defined input and output data, as well as for the complexity and rigorousness of the algorithm itself.
In order to prepare the data for the analysis, we had to binarize the categorical variables. This process included creating dummy variables for the columns InternetService, Contract, and PaymentMethod. We separated our processed data into training and testing sets based on the 80/20 rule.
Our hyperparameter tuning for finding out the optimal number of hidden nodes and weight decay consisted of a grid search and a 10-fold cross-validation repeated 10 times. For the grid, we are considering all sizes from 1 to 7, with an associated decay between 0 and 0.1, increasing by 0.01 for each iteration. Although these chosen values lead to a very high computational complexity, they also generate a more finely tuned and optimized model.

```
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were size = 1 and decay = 0.09.
 1     0.09   0.7958065  0.4509200
```
(Fig. 5)

As we can see from the output of our trained model in Figure 5, the optimal ANN model consists of exactly one hidden node, with a decay parameter of 0.09. The following neural network visualization applies.



(Fig. 6)

We then proceed to predict the Churn classification for both the training and testing datasets. In order to analyze the performance and goodness of fit of our model, we display the confusion matrix statistics for both datasets, using the class “Yes” as the positive value.

	Training	Testing
Accuracy	8.049645e-01	0.7559242
Kappa	4.751318e-01	0.3686489
AccuracyLower	7.852672e-01	0.7120453
AccuracyUpper	8.235980e-01	0.7961726
AccuracyNull	7.346336e-01	0.7345972
AccuracyPValue	8.245686e-12	0.1746104
McnemarPValue	3.460584e-04	0.8437760

(Fig. 7)

As we expected from the nature of the analysis, our Artificial Neural Network model appears to be overfitting based on the significantly higher Accuracy and Kappa values. We will also analyze the performance of the model by class.

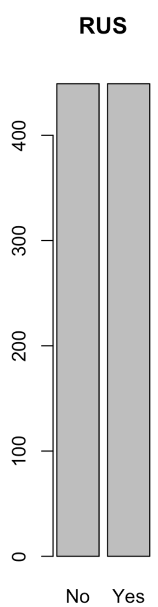
As we can see from Figure 8, most of the Accuracy is accounted for by the correct classification of the customers predicted to not leave the company. However, we are most interested in the value of Sensitivity. Judging by this, our ANN model is underperforming on both the training and testing sets.

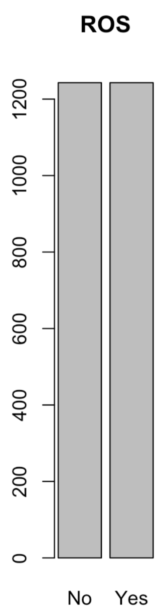
As mentioned in the data description section, the existing class imbalance might be responsible for our model’s inability to accurately predict if a customer will churn or not.

	Training	Testing
Sensitivity	0.5590200	0.5267857
Specificity	0.8938053	0.8387097
Pos Pred Value	0.6553525	0.5412844
Neg Pred Value	0.8487395	0.8306709
Precision	0.6553525	0.5412844
Recall	0.5590200	0.5267857
F1	0.6033654	0.5339367
Prevalence	0.2653664	0.2654028
Detection Rate	0.1483452	0.1398104
Detection Prevalence	0.2263593	0.2582938
Balanced Accuracy	0.7264127	0.6827477

(Fig. 8)

We address our model's shortcomings by training the ANN with both a random undersampled and oversampled dataset. The plots for the updated class distributions are present in Figures 9 and 10. For the undersampled model, the optimal number of hidden nodes is still 1, with a decay of 0.01. For the oversampled model, the optimal number of hidden nodes is 7, with a decay of 0. The following figures represent the performance and goodness of fit outputs for the two datasets. (undersampled first, oversampled second)





	Training	Testing
Accuracy	7.559102e-01	7.203791e-01
Kappa	4.668707e-01	3.846572e-01
AccuracyLower	7.347141e-01	6.749275e-01
AccuracyUpper	7.762148e-01	7.627094e-01
AccuracyNull	7.346336e-01	7.345972e-01
AccuracyPValue	2.456815e-02	7.644700e-01
McnemarPValue	3.286325e-33	1.543651e-07

(Fig. 11)

	Training	Testing
Accuracy	8.008274e-01	7.061611e-01
Kappa	5.617914e-01	3.371504e-01
AccuracyLower	7.809875e-01	6.601751e-01
AccuracyUpper	8.196180e-01	7.492287e-01
AccuracyNull	7.346336e-01	7.345972e-01
AccuracyPValue	1.288490e-10	9.148020e-01
McnemarPValue	1.305836e-36	2.435183e-05

(Fig. 9) (Fig.10) (Fig. 12)

Although the oversampled model presents even more overfitting, we can see a sizable improvement for the undersampled model when it comes to goodness of fit and being balanced.

We will now compare the testing performance of the three models we created overall and by class. (overall first, by class second) The same comparison for the training dataset can be found in Figures 15 and 16.

	Base	Under	Over
Accuracy	0.7559242	7.203791e-01	7.061611e-01
Kappa	0.3686489	3.846572e-01	3.371504e-01
AccuracyLower	0.7120453	6.749275e-01	6.601751e-01
AccuracyUpper	0.7961726	7.627094e-01	7.492287e-01
AccuracyNull	0.7345972	7.345972e-01	7.345972e-01
AccuracyPValue	0.1746104	7.644700e-01	9.148020e-01
McnemarPValue	0.8437760	1.543651e-07	2.435183e-05

	Base	Under	Over
Sensitivity	0.5267857	0.7321429	0.6607143
Specificity	0.8387097	0.7161290	0.7225806
Pos Pred Value	0.5412844	0.4823529	0.4625000
Neg Pred Value	0.8306709	0.8809524	0.8549618
Precision	0.5412844	0.4823529	0.4625000
Recall	0.5267857	0.7321429	0.6607143
F1	0.5339367	0.5815603	0.5441176
Prevalence	0.2654028	0.2654028	0.2654028
Detection Rate	0.1398104	0.1943128	0.1753555
Detection Prevalence	0.2582938	0.4028436	0.3791469
Balanced Accuracy	0.6827477	0.7241359	0.6916475

(Fig. 13)

	Base	Under	Over
Accuracy	8.049645e-01	7.559102e-01	8.008274e-01
Kappa	4.751318e-01	4.668707e-01	5.617914e-01
AccuracyLower	7.852672e-01	7.347141e-01	7.809875e-01
AccuracyUpper	8.235980e-01	7.762148e-01	8.196180e-01
AccuracyNull	7.346336e-01	7.346336e-01	7.346336e-01
AccuracyPValue	8.245686e-12	2.456815e-02	1.288490e-10
McnemarPValue	3.460584e-04	3.286325e-33	1.305836e-36

(Fig. 14)

(Fig. 15)

	Base	Under	Over
Sensitivity	0.5590200	0.8129176	0.8841871
Specificity	0.8938053	0.7353178	0.7707160
Pos Pred Value	0.6553525	0.5259366	0.5821114
Neg Pred Value	0.8487395	0.9158317	0.9485149
Precision	0.6553525	0.5259366	0.5821114
Recall	0.5590200	0.8129176	0.8841871
F1	0.6033654	0.6386702	0.7020336
Prevalence	0.2653664	0.2653664	0.2653664
Detection Rate	0.1483452	0.2157210	0.2346336
Detection Prevalence	0.2263593	0.4101655	0.4030733
Balanced Accuracy	0.7264127	0.7741177	0.8274515

(Fig. 16)

The oversampled model is the worst performing out of the 3. The base and undersampled models have comparable performances, with a small trade-off between Accuracy, which is higher on the base model, and a slightly higher Kappa value and better goodness of fit on the undersampled model. With this being said, the goal of the company is to maximize Sensitivity. Towards this goal, the ANN built on the undersampled dataset far outperforms the base model and even the oversampled one. As such, the Artificial Neural Network trained on the undersampled dataset should be used for predicting future customer churn from the 3 presented models.

Ensemble Methods - Random Forest

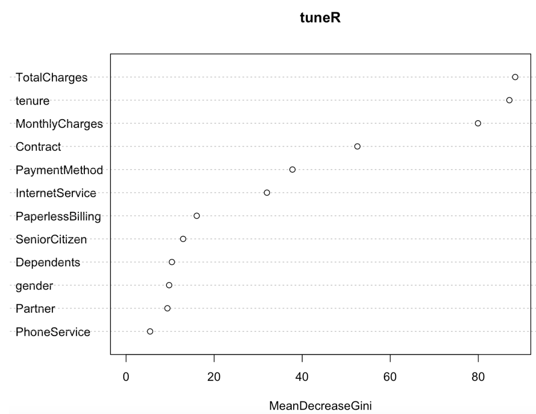
The second analysis method we decided to pursue is Ensemble Methods. Specifically, we will be using Random Forest to predict the customers who will leave the company. We chose this method because of the improved performance, generalizability, stability of classifiers, and reduction in prediction variance of ensemble methods. What's more, we also wanted to de-correlate the trees, hence why we chose Random Forest over Bagging.

Because of the nature of Decision Trees, no further data preparation is needed to commence our model training. The data subsets are split on an 85/15 rule.

We will move straight to the hyperparameter tuning of our base RF model. We will tune the number of variables to randomly sample as potential variables to split on. We chose 500 as the number of trees in the forest.

```
Call:
 randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1])
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 20.13%
Confusion matrix:
      No Yes class.error
No  1204 117  0.08856927
Yes  245 232  0.51362683
```
(Fig. 17)

Our tuned model has 2 variables tried at each split. The plot of predictor importance is as follows. As we can see, the most important variables for our classification are TotalCharges, tenure, and MonthlyCharges (in this order), followed by Contract at a significant distance from the previous.
We move on to predicting the classification for the training and testing datasets with "Yes" as the positive value. The following are the performance and goodness of fit statistics.



(Fig. 18)

	Training	Testing
Accuracy	7.986652e-01	0.8069620253
Kappa	4.351025e-01	0.4442394188
AccuracyLower	7.793713e-01	0.7590598530
AccuracyUpper	8.169793e-01	0.8490029415
AccuracyNull	7.347052e-01	0.7341772152
AccuracyPValue	1.570549e-10	0.0015908719
McnemarPValue	2.472797e-11	0.0003370361

	Training	Testing
Sensitivity	0.4863732	0.4642857
Specificity	0.9114307	0.9310345
Pos Pred Value	0.6647564	0.7090909
Neg Pred Value	0.8309179	0.8275862
Precision	0.6647564	0.7090909
Recall	0.4863732	0.4642857
F1	0.5617433	0.5611511
Prevalence	0.2652948	0.2658228
Detection Rate	0.1290323	0.1234177
Detection Prevalence	0.1941046	0.1740506
Balanced Accuracy	0.6989019	0.6976601

(Fig. 19) (Fig. 20)

Our model displays a balanced goodness of fit, with almost exact Accuracy and Kappa values for both training and testing data. The performance of the Random Forest model is also significantly

better than that of the Artificial Neural Network one for the testing set. However, once again, the high Accuracy results from our model's ability to correctly predict the customer who will stay with the company. In fact, for both training and testing data, our model has a Specificity below 0.5, which means that it is predicting the majority of the positive class wrong.
In order to account for this, we will once again train the model on both a random undersampled and oversampled dataset. The number of variables tried at each split in hyperparameter tuning increased to 3 for the undersampled model, and to 6 for the oversampled model. The following figures present the performance and goodness of fit statistics of the 2 models. (undersampled first, oversampled second)

	Training	Testing
Accuracy	7.777778e-01	7.405063e-01
Kappa	5.555556e-01	4.266242e-01
AccuracyLower	7.500361e-01	6.884756e-01
AccuracyUpper	8.037927e-01	7.879648e-01
AccuracyNull	5.000000e-01	7.341772e-01
AccuracyPValue	8.341512e-70	4.280482e-01
McnemarPValue	2.429824e-01	5.963125e-06

(Fig. 21)

	Training	Testing
Accuracy	9.091597e-01	0.78164557
Kappa	8.183195e-01	0.42975207
AccuracyLower	8.975539e-01	0.73197474
AccuracyUpper	9.198493e-01	0.82594906
AccuracyNull	5.000000e-01	0.73417722
AccuracyPValue	0.000000e+00	0.03052345
McnemarPValue	3.125712e-30	0.63013033

(Fig. 22)

Both models saw a decrease in the goodness of fit, although the oversampled model's training Kappa value is 2 times that of the testing value. We will next compare the testing performance of all 3 Random Forest models created. The same comparison for the training data can be found in Figures 25 and 26. (overall first, by class second)

	Base	Under	Over
Accuracy	0.8069620253	7.405063e-01	0.78164557
Kappa	0.4442394188	4.266242e-01	0.42975207
AccuracyLower	0.7590598530	6.884756e-01	0.73197474
AccuracyUpper	0.8490029415	7.879648e-01	0.82594906
AccuracyNull	0.7341772152	7.341772e-01	0.73417722
AccuracyPValue	0.0015908719	4.280482e-01	0.03052345
McnemarPValue	0.0003370361	5.963125e-06	0.63013033

(Fig. 23)

	Base	Under	Over
Sensitivity	0.4642857	0.7619048	0.5595238
Specificity	0.9310345	0.7327586	0.8620690
Pos Pred Value	0.7090909	0.5079365	0.5949367
Neg Pred Value	0.8275862	0.8947368	0.8438819
Precision	0.7090909	0.5079365	0.5949367
Recall	0.4642857	0.7619048	0.5595238
F1	0.5611511	0.6095238	0.5766871
Prevalence	0.2658228	0.2658228	0.2658228
Detection Rate	0.1234177	0.2025316	0.1487342
Detection Prevalence	0.1740506	0.3987342	0.2500000
Balanced Accuracy	0.6976601	0.7473317	0.7107964

(Fig. 24)

	Base	Under	Over
Accuracy	7.986652e-01	7.777778e-01	9.091597e-01
Kappa	4.351025e-01	5.555556e-01	8.183195e-01
AccuracyLower	7.793713e-01	7.500361e-01	8.975539e-01
AccuracyUpper	8.169793e-01	8.037927e-01	9.198493e-01
AccuracyNull	7.347052e-01	5.000000e-01	5.000000e-01
AccuracyPValue	1.570549e-10	8.341512e-70	0.000000e+00
McnemarPValue	2.472797e-11	2.429824e-01	3.125712e-30

(Fig. 25)

	Base	Under	Over
Sensitivity	0.4863732	0.7966457	0.9765329
Specificity	0.9114307	0.7589099	0.8417865
Pos Pred Value	0.6647564	0.7676768	0.8605737
Neg Pred Value	0.8309179	0.7886710	0.9728784
Precision	0.6647564	0.7676768	0.8605737
Recall	0.4863732	0.7966457	0.9765329
F1	0.5617433	0.7818930	0.9148936
Prevalence	0.2652948	0.5000000	0.5000000
Detection Rate	0.1290323	0.3983229	0.4882665
Detection Prevalence	0.1941046	0.5188679	0.5673732
Balanced Accuracy	0.6989019	0.7777778	0.9091597

v(Fig. 26)

This time around, the best model for the purposes of our analysis is not as obvious as for the ANN model. The best performing model still appears to be the base model, with the highest Accuracy and Kappa values. However, the difference between all 3 models is not significant enough to discount either the undersampled or oversampled models. The difference becomes more apparent when we consider the goal of the company, which is to maximize Sensitivity. The RF model built on the undersampled training data is far superior to the others when it comes to correctly predict the customers who will leave the company. Factoring both the comparable overall performance and the exponentially higher Sensitivity, the undersampled model is again the best choice for predicting future customer churn out of the 3 Ensemble Method models.

Comparing the Artificial Neural Network and Random Forest models

The final step in our analysis and interpretation is to choose the best model out of the 6 ones we created. The following figures contain the performance statistics comparison between ANN and RF for the models built on the base, undersampled, and oversampled testing data. The same comparison for the training data can be found in Figures 30, 31, and 32.

	ANN	RF
Accuracy	0.7559242	0.8069620253
Kappa	0.3686489	0.4442394188
AccuracyLower	0.7120453	0.7590598530
AccuracyUpper	0.7961726	0.8490029415
AccuracyNull	0.7345972	0.7341772152
AccuracyPValue	0.1746104	0.0015908719
McnemarPValue	0.8437760	0.0003370361

	ANN_US	RF_US
Accuracy	7.203791e-01	7.405063e-01
Kappa	3.846572e-01	4.266242e-01
AccuracyLower	6.749275e-01	6.884756e-01
AccuracyUpper	7.627094e-01	7.879648e-01
AccuracyNull	7.345972e-01	7.341772e-01
AccuracyPValue	7.644700e-01	4.280482e-01
McnemarPValue	1.543651e-07	5.963125e-06

	ANN_OS	RF_OS
Accuracy	7.061611e-01	0.78164557
Kappa	3.371504e-01	0.42975207
AccuracyLower	6.601751e-01	0.73197474
AccuracyUpper	7.492287e-01	0.82594906
AccuracyNull	7.345972e-01	0.73417722
AccuracyPValue	9.148020e-01	0.03052345
McnemarPValue	2.435183e-05	0.63013033

	ANN	RF
Sensitivity	0.5267857	0.4642857
Specificity	0.8387097	0.9310345
Pos Pred Value	0.5412844	0.7090909
Neg Pred Value	0.8306709	0.8275862
Precision	0.5412844	0.7090909
Recall	0.5267857	0.4642857
F1	0.5339367	0.5611511
Prevalence	0.2654028	0.2658228
Detection Rate	0.1398104	0.1234177
Detection Prevalence	0.2582938	0.1740506
Balanced Accuracy	0.6827477	0.6976601

(Fig. 27)

	ANN_US	RF_US
Sensitivity	0.7321429	0.7619048
Specificity	0.7161290	0.7327586
Pos Pred Value	0.4823529	0.5079365
Neg Pred Value	0.8809524	0.8947368
Precision	0.4823529	0.5079365
Recall	0.7321429	0.7619048
F1	0.5815603	0.6095238
Prevalence	0.2654028	0.2658228
Detection Rate	0.1943128	0.2025316
Detection Prevalence	0.4028436	0.3987342
Balanced Accuracy	0.7241359	0.7473317

(Fig.28)

	ANN_OS	RF_OS
Sensitivity	0.6607143	0.5595238
Specificity	0.7225806	0.8620690
Pos Pred Value	0.4625000	0.5949367
Neg Pred Value	0.8549618	0.8438819
Precision	0.4625000	0.5949367
Recall	0.6607143	0.5595238
F1	0.5441176	0.5766871
Prevalence	0.2654028	0.2658228
Detection Rate	0.1753555	0.1487342
Detection Prevalence	0.3791469	0.2500000
Balanced Accuracy	0.6916475	0.7107964

(Fig. 29)

	ANN	RF
Accuracy	8.049645e-01	7.986652e-01
Kappa	4.751318e-01	4.351025e-01
AccuracyLower	7.852672e-01	7.793713e-01
AccuracyUpper	8.235980e-01	8.169793e-01
AccuracyNull	7.346336e-01	7.347052e-01
AccuracyPValue	8.245686e-12	1.570549e-10
McnemarPValue	3.460584e-04	2.472797e-11

	ANN_US	RF_US
Accuracy	7.559102e-01	7.777778e-01
Kappa	4.668707e-01	5.555556e-01
AccuracyLower	7.347141e-01	7.500361e-01
AccuracyUpper	7.762148e-01	8.037927e-01
AccuracyNull	7.346336e-01	5.000000e-01
AccuracyPValue	2.456815e-02	8.341512e-70
McnemarPValue	3.286325e-33	2.429824e-01

	ANN_OS	RF_OS
Accuracy	8.008274e-01	9.091597e-01
Kappa	5.617914e-01	8.183195e-01
AccuracyLower	7.809875e-01	8.975539e-01
AccuracyUpper	8.196180e-01	9.198493e-01
AccuracyNull	7.346336e-01	5.000000e-01
AccuracyPValue	1.288490e-10	0.000000e+00
McnemarPValue	1.305836e-36	3.125712e-30

	ANN	RF
Sensitivity	0.5590200	0.4863732
Specificity	0.8938053	0.9114307
Pos Pred Value	0.6553525	0.6647564
Neg Pred Value	0.8487395	0.8309179
Precision	0.6553525	0.6647564
Recall	0.5590200	0.4863732
F1	0.6033654	0.5617433
Prevalence	0.2653664	0.2652948
Detection Rate	0.1483452	0.1290323
Detection Prevalence	0.2263593	0.1941046
Balanced Accuracy	0.7264127	0.6989019

(Fig. 30)

	ANN_US	RF_US
Sensitivity	0.8129176	0.7966457
Specificity	0.7353178	0.7589099
Pos Pred Value	0.5259366	0.7676768
Neg Pred Value	0.9158317	0.7886710
Precision	0.5259366	0.7676768
Recall	0.8129176	0.7966457
F1	0.6386702	0.7818930
Prevalence	0.2653664	0.5000000
Detection Rate	0.2157210	0.3983229
Detection Prevalence	0.4101655	0.5188679
Balanced Accuracy	0.7741177	0.7777778

(Fig. 31)

	ANN_OS	RF_OS
Sensitivity	0.8841871	0.9765329
Specificity	0.7707160	0.8417865
Pos Pred Value	0.5821114	0.8605737
Neg Pred Value	0.9485149	0.9728784
Precision	0.5821114	0.8605737
Recall	0.8841871	0.9765329
F1	0.7020336	0.9148936
Prevalence	0.2653664	0.5000000
Detection Rate	0.2346336	0.4882665
Detection Prevalence	0.4030733	0.5673732
Balanced Accuracy	0.8274515	0.9091597

(Fig. 32)

Evidently, the best model will be either the ANN built on undersampled training data or the RF built on undersampled training data. Looking at the comparison between the two, we can safely say that the Random Forest model is the superior one, as the Accuracy, Kappa, and Sensitivity values are all slightly higher. As such, we can conclude that out of the 2 analysis methods we tried on the dataset, the undersampled RF model should be used to predict if future customers will leave the company.

Conclusion

Based on our analysis of TKT's customer database, we were able to determine the order of importance for predictor variables pertaining to customer churn. Total charges, tenure, and monthly charges had the highest importance in our model, as shown in Figure 18. With this pertinent information, we then were able to determine that the undersampled RF model would have the best fit. The Accuracy, Kappa, and Sensitivity values of the undersampled RF model validated that it ultimately was the preferred model in this use case. Moving forward, we recommend that the management of TKT strongly consider using our undersampled RF model to predict future customer churn rates. With this as a tool, TKT will be better equipped to determine which of their customers may leave and allow TKT to take appropriate, preventative measures to try and keep them as customers to maximize customer retention.