

Telecommunication Customer Attrition

Ruben Brionez Jr

Bellevue University

DSC630 - Predictive Analytics

Andrew Hua

August 10, 2024

Telecommunication Customer Attrition

The telecommunication industry is an ever growing and changing industry in the United States. It has become one of the most vital industries that affects people's everyday lives. The telecommunications industry brings internet, voice, and cable T.V. services to millions of residential and commercial customers across the United States. One of the challenges that this industry faces is customer attrition. With many different providers, competition and promotions, there tend to be fluctuations in the number of customers a telecommunications company provides service to.

"The churn rate, also known as the rate of attrition or customer churn, is the rate at which customers stop doing business with an entity. It is most commonly expressed as the percentage of service subscribers who discontinue their subscriptions within a given time period" (Cheng, M., & Kvilhaug, S., 2024). The underlying goal of exploring and analyzing customer attrition is finding patterns and inferring or exploring possible causes as to why customers are leaving. Having a better understanding of customer churn allows a company to update their approach to keeping customers and improving revenue and sales.

Data Selection

For this project I will be analyzing the data and using it to predict the churn for a fictitious telecommunications company. The initial dataset for this project was sourced from Kaggle.com and is titled "WA_Fn-UseC_-Telco-Customer-Churn.csv" (BlastChar, 2018). The selected data set provides the data required to analyze and predict churn for a fictitious telecommunications company. It includes 7043 rows and 21 columns, or features, for use in analyzing and predicting churn. The rows of the data set represent individual customers, while the features of the data set include descriptive data of the customer, such as age, gender, tenure, and types of service.

Although this is the initial dataset, there may be more data needed as the project progresses. The process of cleaning and engineering may reduce the dataset to an undesirably small size so as I

begin working with the data and project it may be necessary to bring in additional data. Additional data may also be required as model selection and testing begin to take place.

Model Selection

This project will require a model to predict the customer churn for the telecommunication company. The objective after cleaning and preparing the data set for modeling is to use the data set for a linear regression model. Since the target will be a single feature, churn, a simple linear regression model should be sufficient to use a prediction model. However, multiple models may be evaluated with various features to be sure the best model is selected to represent the churn predictions.

The other model that may be considered is a variation of the linear regression model, a multiple regression model. With this model, multiple features could be evaluated together to get a better understanding of how multiple features will affect the target of the model. Both models will probably be included in the final analysis of the project for comparison.

Result Evaluation

Evaluation of the model will vary depending on the model used. The linear regression model will be evaluated using the RMSE and R-squared methods to check the predictions against actual values. A score using the R-squared metric of 0.70 or above is generally regarded as an indication of a good linear regression model.

When reviewing the results of the analysis of customer attrition, I want to find the factors that appear to correlate to an increase in higher customer attrition rates. Once the most influential factors are discovered, exploration of the causes can begin, and some inferences can be made as to why the factors are contributing to customer attrition.

Learning Objectives

While working through this project I have multiple learning objectives and goals that I would like to meet:

- A better understanding of how customer attrition is calculated and evaluated within the telecommunication industry.
- A thorough understanding of how machine learning models can be utilized to in the telecommunication industry to predict and analyze customer attrition
- An understanding of what to do with the insights gained from predictive models and analysis regarding customer attrition.

Having worked for ten years in the telecommunications industry, this is a great opportunity to expand on my industry knowledge and gain further insights into a part of the industry I have a had little experience with.

Risks and Ethical Considerations

There are multiple risks associated with the project. One of the biggest risks that I currently see is the risk of a small data set. With only roughly 7000 rows of customer data and not having cleaned and prepared the data yet, I have a concern that the data set might end up too small. Generally, from my experience in the telecommunications industry, service providers have many more customers than 7000. It would be ideal to have a larger data set to work with for the project.

An ethical consideration of the project is how the analysis and predictions may be used. Will the results of the predictions and analysis impact certain demographics' ability to attain high speed internet and cable services? Will the results of the analysis give a certain group better financial deals than other groups? These issues, at a minimum, should be discussed and evaluated.

Contingency Planning

Contingency planning is something that should always be considered in any project, whether it is an academic project or a professional project. There should always be additional options to consider when completing a project. I have come up with a few for this project:

- **Finding Additional Data:** If after cleaning and preparing the data, the original data set becomes too small. Alternative sets of data will be sourced to supplement the existing data.
- **Model Selection and Target:** If after selecting a model, the results are not desirable or do not appear to make sense for the target, the target and features may need to be adjusted. This may change the scope of the project.
- **Alternative Project:** If for some reason there are issues and barriers that cannot be resolved in a timely manner, I have a backup data set and project in mind. The biggest concern in using the backup project is coming up with a viable business problem or business question to justify the data set.

Preliminary Analysis

Will I be able to answer my original questions with the original data set

The data set originally selected was already developed to answer the types of questions asked. This was a driving factor in the decision to use the data set. The data set originally selected provided features that included customer demographics, different levels of service for customers, as well as a churn feature, which is our target for the predictive analysis. I remain confident that the data set provided will be able to answer the original questions.

Visualization that are useful for explaining the data

During the preliminary analysis, histograms have been especially useful in visualizing the different customer demographic features of the data set. Histograms are a quick easy way to visualize how many times certain values of features appear in the data set. Another useful visualization I found during the preliminary analysis was a box plot. Using a boxplot allows the visualization of the summary of numeric features in the data set. The box plot visualization is also exceedingly helpful at identifying outliers in the numeric features of the dataset.

Does the data or driving questions need to be adjusted

After the preliminary analysis of the data set, the data and driving questions do not need to be changed or adjusted. As discussed in a prior section, the data was created to represent a fictitious telecommunication company in order to study and evaluate models for the prediction of churn. The data is specifically built for this type of analysis and should allow for a good example of real-world model creation and implementation of predictive analytics.

Do the model and evaluation choices need to be modified

At this point, after the preliminary analysis, I do believe the model and evaluation choices may need to be modified. The churn data type appears to be categorical, and more specifically, binary values. The linear regression model initially picked would not be a good fit for this type of data, instead, a logistical regression model may be a better fit for the binary target values.

Changing the model type would also create a need to change the evaluation of the model. For a classification model like logistical regression, a confusion matrix may be a good fit for evaluation. Accuracy, recall and F1 scores would also been good evaluation metrics for the classification model. I plan on using various python libraries, like scikit, to run these evaluations and metric scores against the selected model.

Are original expectations still reasonable

After the realization that the model and evaluation may need to be changed, the original expectations remain reasonable and unchanged. Using the original data set with a modified model and evaluation should still allow for accurate predictive analysis of customer attrition, or churn. Even though the model will change, the new model selected should prove even better after this preliminary analysis.

Finalizing Results

At the conclusion of Milestone 4 of this project several steps will be completed. Those steps include the preparation and preprocessing of the data set to create a working model, the final selection and implementation of the model, and finally, the evaluation of the selected model.

Data Preparation

The data set selected for this project, as mentioned in previous milestones, was specifically created for churn analysis. This resulted in starting with an unusually clean data set that did not contain missing values. However, there were some cleaning steps required before pre-processing the data. Specifically, I took note of the different data types within the data frame and noticed that at least one column, the 'TotalCharges' column, was an incorrect type, this can be seen below in Figure 1 at index 19. The apply method was used to accomplish this and the code can be seen in Figure 2.

The above-mentioned step was the only step that was taken to clean the data set. The remaining steps involved the pre-processing of the data to get ready for the model, as well as the actual model implementation and evaluation.

Figure 1*Data Types of Features in the Data Frame*

```
[8]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   customerID            7043 non-null   object
1   gender                7043 non-null   object
2   SeniorCitizen         7043 non-null   int64
3   Partner               7043 non-null   object
4   Dependents            7043 non-null   object
5   tenure                7043 non-null   int64
6   PhoneService          7043 non-null   object
7   MultipleLines         7043 non-null   object
8   InternetService       7043 non-null   object
9   OnlineSecurity        7043 non-null   object
10  OnlineBackup          7043 non-null   object
11  DeviceProtection      7043 non-null   object
12  TechSupport           7043 non-null   object
13  StreamingTV           7043 non-null   object
14  StreamingMovies       7043 non-null   object
15  Contract              7043 non-null   object
16  PaperlessBilling      7043 non-null   object
17  PaymentMethod         7043 non-null   object
18  MonthlyCharges        7043 non-null   float64
19  TotalCharges          7043 non-null   object
20  Churn                 7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Note: This figure illustrates that the ‘TotalCharges’ feature in the data frame was an object data type.

Since this feature is a monetary value, it needed to be converted to a float64 data type.

Figure 2*Code Snippet for Converting Data Type*

```
[12]: # converting the totalcharges column from object to float64, this will match the monthlycharges column
df['TotalCharges'] = df['TotalCharges'].apply(lambda x: pd.to_numeric(x, errors='coerce')).dropna()
```

Note: This figure illustrates the code used to change the data type of the feature ‘TotalCharges’.

Pre-Processing

The pre-processing of the data frame started with the separation of the categorical features from the numerical features. This was done so the categorical features can be converted to numerical features and used in the machine learning model. In order to convert the features, the scikit-learn library for pre-processing was utilized. The `LabelEncoder()` function was used from the scikit-learn pre-processing library was used to convert the categorical features to numeric features. Figure 3 shows the categorical features before while Figure 4 shows the features converted.

Figure 3

Categorical Feature Before Transformation

```
[13]: # reviewing categorical variables from data frame
cat_features = df.drop(['customerID', 'TotalCharges', 'MonthlyCharges', 'SeniorCitizen', 'tenure'], axis=1)
cat_features.head()
```

	gender	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingM
0	Female	Yes	No	No	No phone service	DSL	No	Yes	No	No	No	
1	Male	No	No	Yes	No	DSL	Yes	No	Yes	No	No	
2	Male	No	No	Yes	No	DSL	Yes	Yes	No	No	No	
3	Male	No	No	No	No phone service	DSL	Yes	No	Yes	Yes	No	
4	Female	No	No	Yes	No	Fiber optic	No	No	No	No	No	

Note: Figure 3 illustrates the categorical features before they were transformed using the `LabelEncoder()` function.

Figure 4*Categorical Features After Transformation with Code Snippet*

```
[14]: # importing additional library for pre-processing
from sklearn import preprocessing
```

```
[15]: # creating a label encoder to transform and fit the categorical features
label_encoder = preprocessing.LabelEncoder()
df_cat = cat_features.apply(label_encoder.fit_transform)
df_cat.head()
```

	gender	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingM
0	0	1	0	0	1	0	0	2	0	0	0	
1	1	0	0	1	0	0	2	0	2	0	0	
2	1	0	0	1	0	0	2	2	0	0	0	
3	1	0	0	0	1	0	2	0	2	2	0	
4	0	0	0	1	0	1	0	0	0	0	0	

Note: Figure 4 illustrates

The second step in the pre-processing of the data frame was to re-combine the separated categorical features with the original numerical features. Once re-combined, the data frame can then be split into training and test sets, this was completed using the scikit-learn “train_test_split” library.

The final step of the pre-processing step was to oversample the training data set. This was done to balance the training set. Since the original data frame was unbalanced in the number of customers that churned versus those that did not. An article from Medium has this to say about over sampling:

For oversampling, we will use SMOTE, which is a widely used technique in classification problems where the minority class is significantly smaller than the majority class. The technique works by selecting an example from the minority class and finding its k nearest neighbors. It then creates new synthetic examples by randomly interpolating the attributes of the selected examples and adding them to the dataset (Santiago, D. 2023.).

This technique should allow the machine learning model to better predict the target feature, the oversampling will only be applied to the training data set. This should allow the model to be accurate and the test set remain true to the original data.

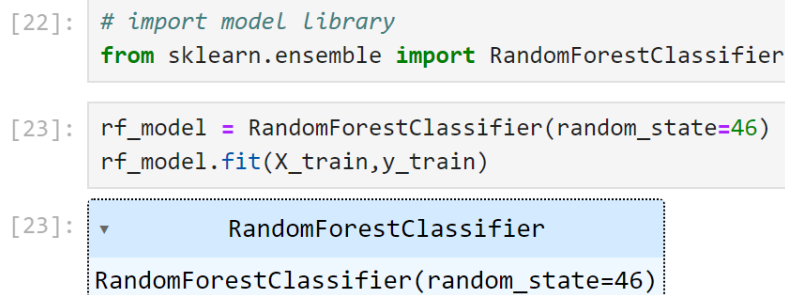
Model Implementation

The model being used for implementation is a random forest model. Throughout the project, the model being chosen has evolved. The original model was to be a linear regression, but the categorical features lead to the possibility of a logistic regression model. Ultimately, with the random forest being a form of decision tree, it seemed the most appropriate.

The model was trained using the training data set that had oversampling applied. The model was implemented using the scikit-learn library. Figure 5 shows the implementation of the model.

Figure 5

Random Forest Implementation



```
[22]: # import model library
      from sklearn.ensemble import RandomForestClassifier

[23]: rf_model = RandomForestClassifier(random_state=46)
      rf_model.fit(X_train,y_train)

[23]: ▼      RandomForestClassifier
      RandomForestClassifier(random_state=46)
```

Note: Figure 5 illustrates the random forest model being implanted on the training set

Evaluation

Once the model was trained with the training data set, the model was then ready for use on the test set. Using the model on the test set allowed for predictions to be created for churn. The model considered all the categorical features to predict the target feature of churn.

To evaluate the model, the “accuracy_score” function was imported from the scikit-learn library. The predictions made using the random forest model were then compared to the actual values from the test data set. The accuracy score function was used on the predictions and actual values to return the

score. The score returned was 0.77, as general estimate anything above 0.70 is considered a good model. Figure 6 shows the implementation of the random forest model on the test data.

Figure 6

Random Forest Model with Test Data

```
[24]: # import library for evaluating model
      from sklearn.metrics import accuracy_score

[25]: predictions = rf_model.predict(X_test)
      print(accuracy_score(predictions,y_test))

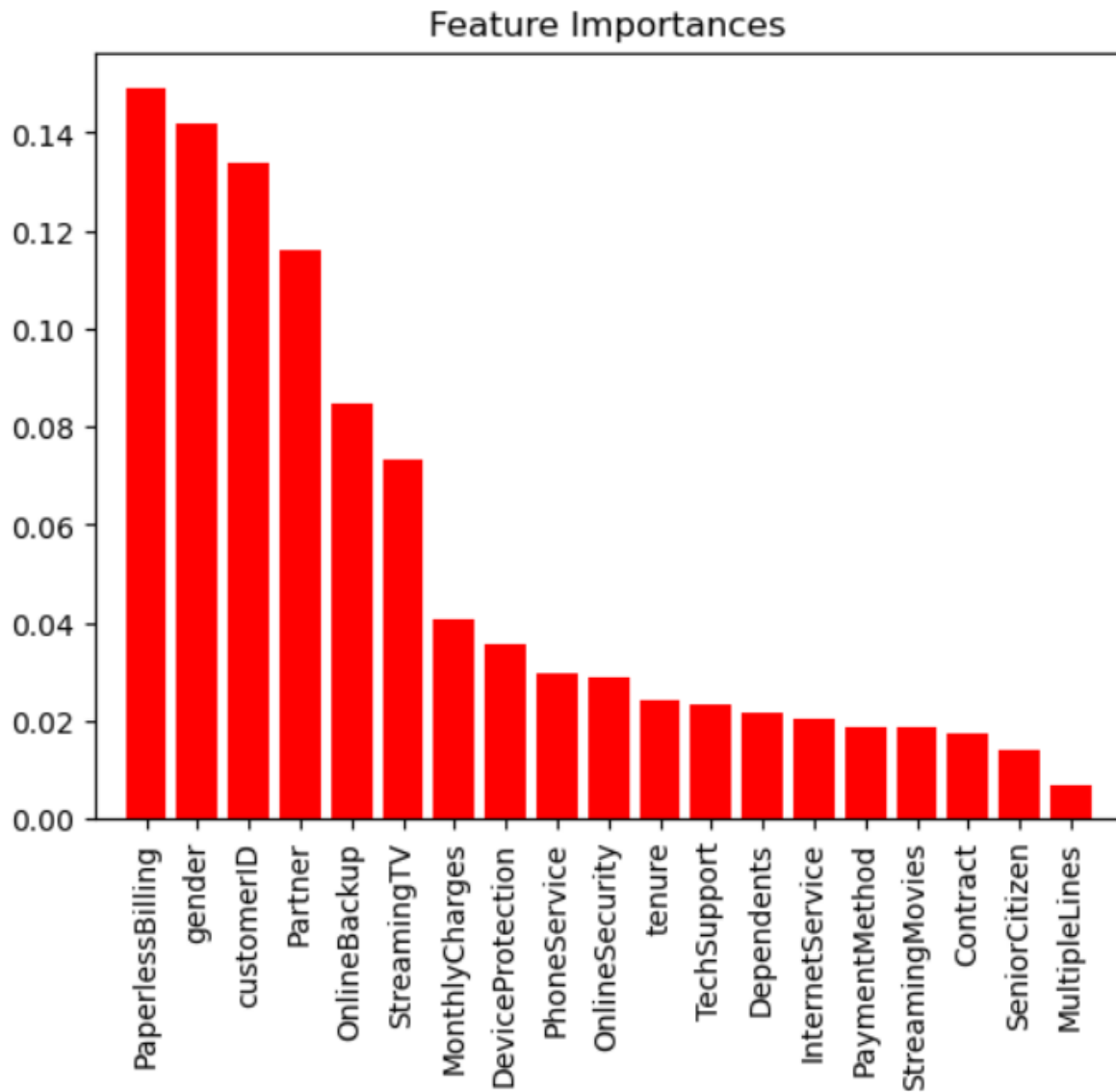
0.7712193020249892
```

Note: Figure 6 illustrates the implementation of the accuracy score function on the predictions versus the actual values.

Another useful tool for evaluating random forest models is feature importance. Feature importance allows us to analyze which features had the largest impact on the random forest model predictions. This in turn allows us to focus efforts on those features to reduce the churn of customers. Feature importance is a pivotal part in understanding the results of the Random Forest model. Without feature importance there is no indication of why the model made the predictions. In Figure 7, the feature importance is displayed in a bar graph.

Figure 7

Bar graph of Feature Importance



Note: Figure 7 illustrates the bar graph of Feature Importance. This is a pivotal part of Random Forest models as it showcased the features that had the largest impacts on the predictions.

Conclusion and Recommendations

The model appears to be functioning accurately based on the accuracy score. The Random Forest model appears to be acceptable at making predictions based on the accuracy score result. Some simple observations during exploratory data analysis can help in providing recommendations on what

may be causing the prediction from the Random Forest model. One thing that was noted during EDA was that when comparing monthly charges to churn, the customers with the higher monthly charges were more likely to churn.

However, with the inclusion of the Feature Importance bar graph we can get a much better understanding of the Random Forest model predictions. The Feature Importance graph showcases the most important features that impacted the model. Based on the graph, we can see that the Paperless Billing feature had the largest impact on the model's predictions, followed by gender.

Gender may not be a feature that we wish to tackle as an issue, but we could focus on paperless billing. It is apparent that paperless billing had the largest impact on the predictions and further analysis tells us that customers with paperless billing were less likely to churn while customers without paperless billing were more likely to churn. The recommendation then would be to increase enrollment in paperless billing to reduce churn.

References

Cheng, M., & Kvilhaug, S. (Eds.). (2024, March 21). *Churn rate: What it means, examples, and calculations*. Investopedia.

<https://www.investopedia.com/terms/c/churnrate.asp>

BlastChar. (2018). *Telco Customer Churn* (WA_Fn-UseC_-Telco-Customer-Churn.csv)[Dataset]. Kaggle.

<https://www.kaggle.com/datasets/blatchar/telco-customer-churn>

Santiago, D. (2023, June 5). *Balancing imbalanced data: Undersampling and oversampling techniques in Python*. Medium.

<https://medium.com/@daniele.santiago/balancing-imbalanced-data-undersampling-and-oversampling-techniques-in-python>