Capstone2 Final Report:

The Introduction

The Problem:

Can I use a data set from the fictional telecommunications company Telco to predict customer churn? If so, can I identify which features have the greatest predictive power in identifying those customers who will leave?

The Why:

As the author and speaker, Jeffery Gitomer, states in the title of his book, “Customer Satisfaction Is Worthless, Customer Loyalty Is Priceless”, having customers that stick with your company is extremely important. That is why it is of great value to a business to be able to identify those customers who will potentially leave the business and why.

For Telco specifically, here is why it matters. The customers who left spent a total of $139,130.85/month with an average of $74.44/month per customer. The customers who are still with Telco spend $316,985.75/month with an average of $61.27/month per customer. This is a 30.5% per month loss due to customer churn! Our best model was able to predict 79% of the customers who would leave. If just half of the 79% the model predicted to leave Telco were convinced to stay, then that would be a savings of $54,956.68/month. This would reduce the per month loss to 18.5%!

The Solution:

After cleaning the data set and conducting extensive exploratory data analysis (EDA) I was able to train a logistic regression model in conjunction with the SMOTE oversampling technique that had an overall accuracy of 0.75 and a recall score on the target feature of 0.79.

If you are interested in understanding more of the discovery process that led to this model and the rejection of many others, please peruse these notebooks. I suggest the order given, but each notebook should be able to stand on its own as well.

1.0_Capstone2_cleaningdata
2.0_Capstone2_EDA
2.1_Capstone2_featureselection
3.0_Capstone2_LogisticRegression_initialassessment
3.1_Capstone2_LogisticRegression_imbalanceddata
3.2_Capstone2_randomforest
3.3_Capstone2_LogReg_and_RFC_usingselectedfeatures
4.0_Capstone2_furtherexploration_year1model

The Process

The Data:

The data set that I will be using was obtained from Kaggle:

https://www.kaggle.com/blastchar/telco-customer-churn

It was originally obtained from an IBM Cognos Analytics sample data set. The IBM data set has numerous other features that were not included in the data set used in this project obtained from the Kaggle source.

https://www.ibm.com/support/knowledgecenter/SSEP7J_11.1.0/com.ibm.swg.ba.cognos.ig_smples.doc/c_telco_dm_sam.html

I downloaded the data set as a csv file from Kaggle. I then loaded the raw csv data file to my GitHub. I then loaded the data as a pandas DataFrame into the Capstone2_cleaningdata notebook. With the data as a pandas DataFrame, I was able to scour it for missing values, anomalous entries, and to make sure the data types of the features were correct.

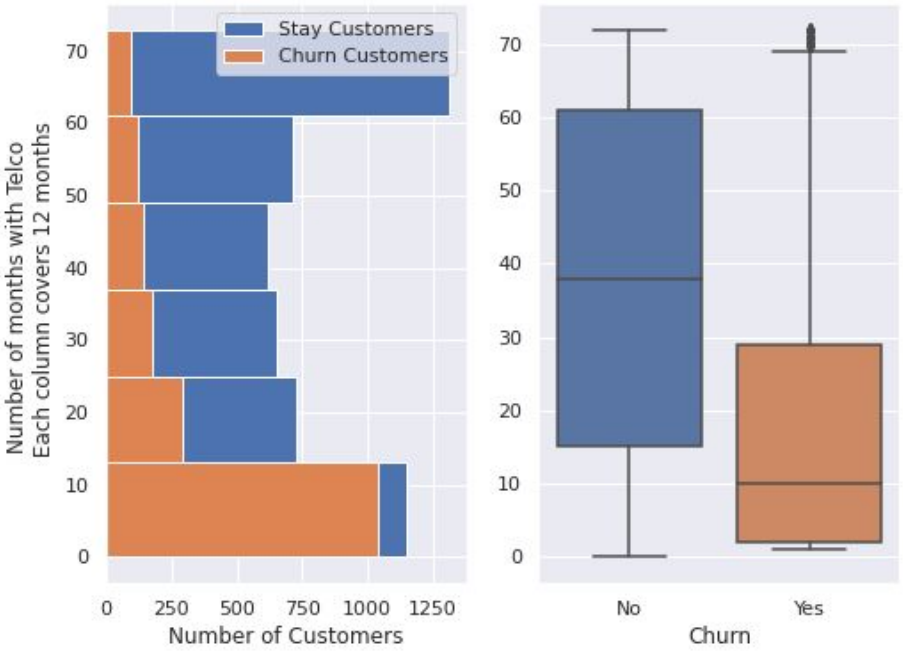
I first noticed an issue when I observed that the 'TotalCharges' feature was listed as an object dtype instead of a numeric dtype (I was expecting a float64 dtype as it's sister column, 'MonthlyCharges' was, or even an int64 dtype). This indicated to me that there were some non-number entries in this feature. The search for missing or NaN entries returned fruitless, so I knew they were not the culprit. I decided to force the situation by using .to_numeric on the troubling column with errors='coerce'. This would cause any non-numeric entry to become a NaN. This worked! By using .isna().any(axis=1) on the DataFrame, I was able to isolate the troublesome rows. There were only 11, so this allowed me to inspect them all visually. Upon close inspection, I noticed that they all had a 'tenure' value of 0. It was at that moment that the story of the incorrect column dtype came into focus.

These rows did not have any 'TotalCharges' because they were all brand new customers who had not been charged anything yet. With this new knowledge in hand, I quickly changed their NaN's in the 'TotalCharges' feature to 0, as that is exactly what they had been charged so far.

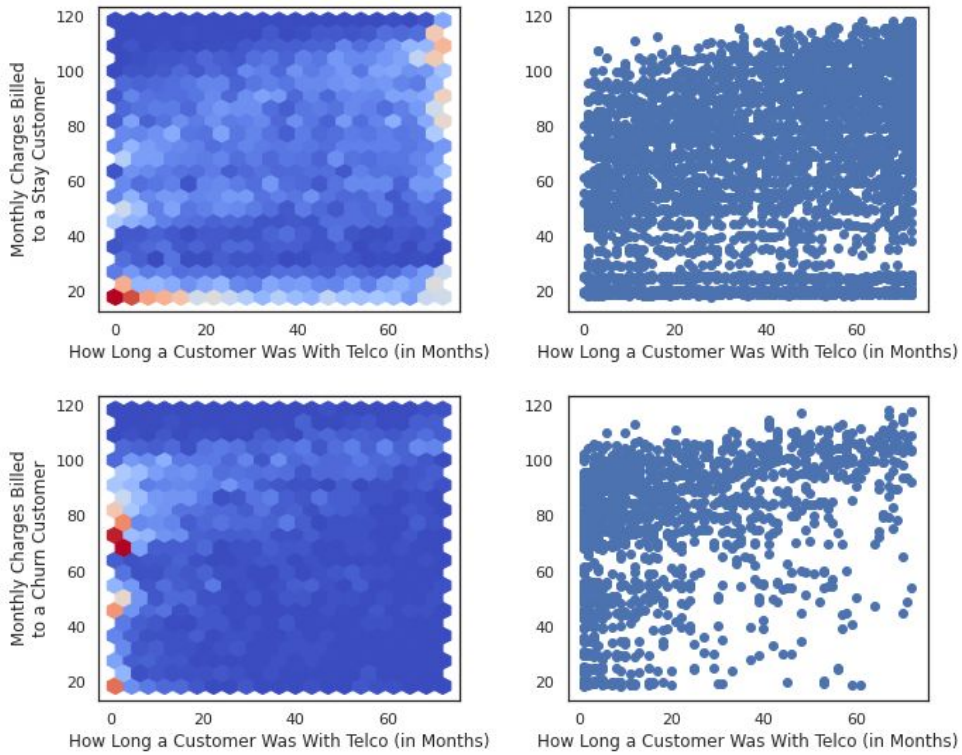
The rest of the data looked clean and ready for some EDA!

EDA:

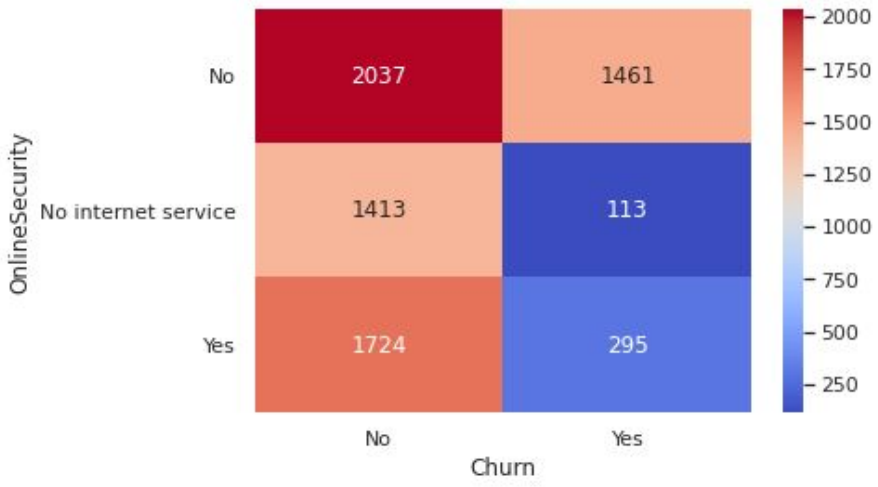
This data set, although not as large or as intricate as others, still held delightful surprises to unearth upon exploration.
One of the first themes to be encountered when exploring the Telco churn data is how many customers left in less than one year of being with Telco. This is easily seen in the chart below. I go on to study this issue more in Capstone2_furtherexploration_year1model notebook.



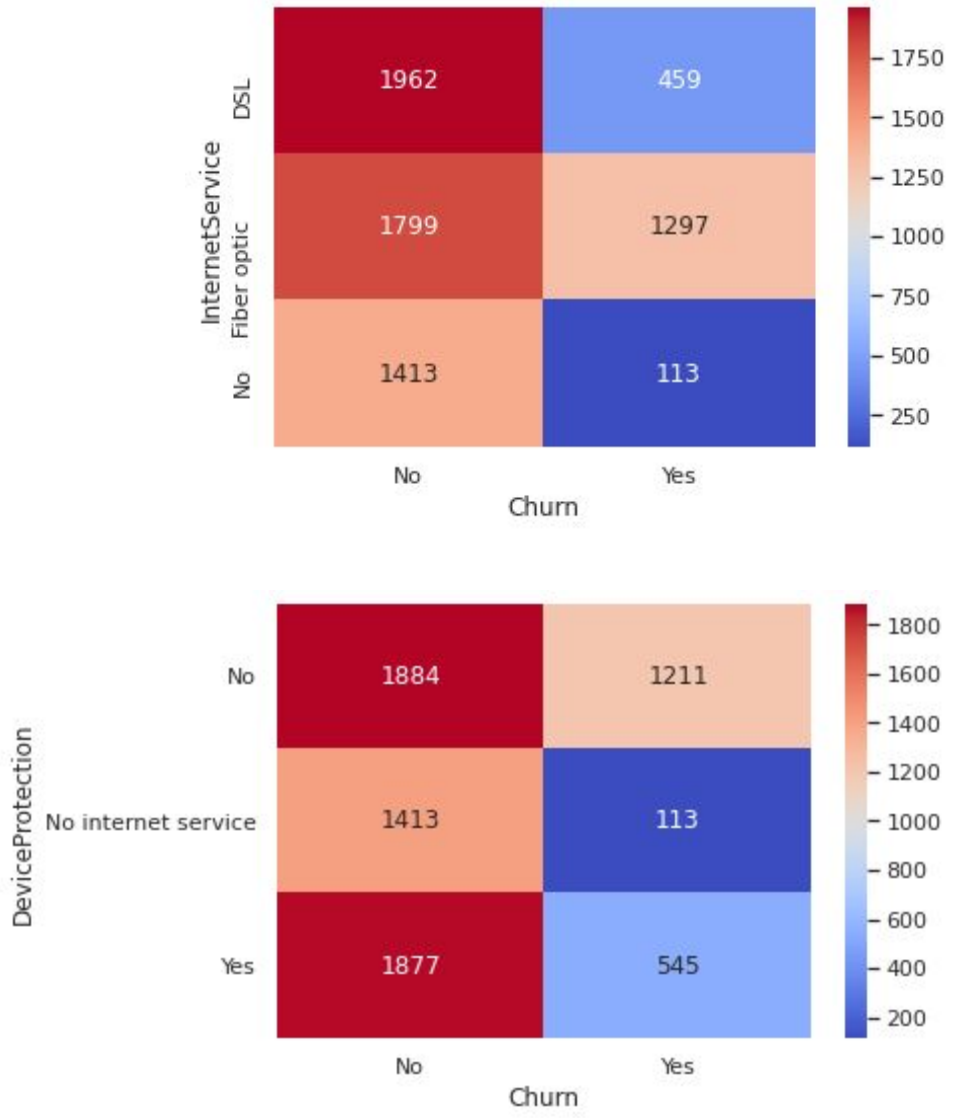
Another theme that stood out was the concentration disparity related to the monthly charges of a customer and the tenure of that customer. This can be seen in this figure.



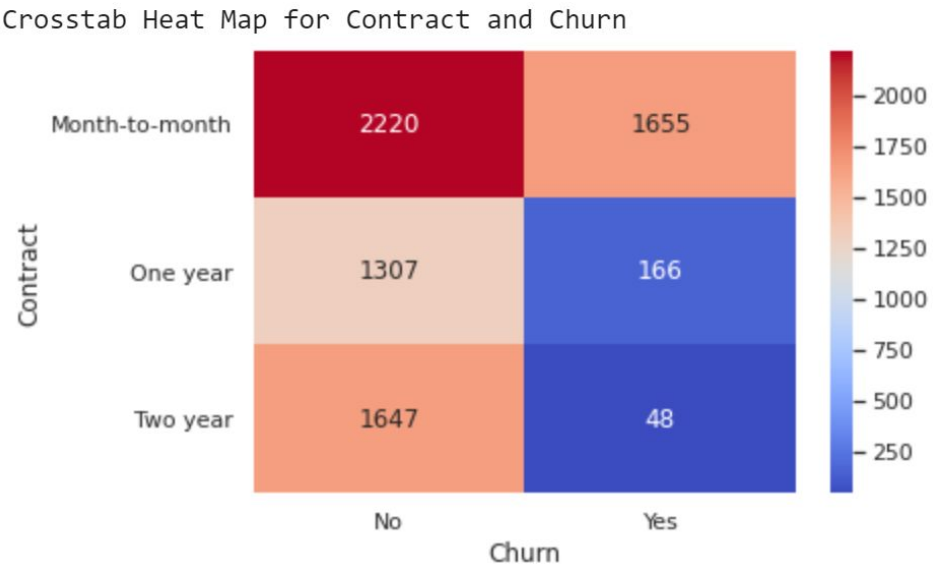
When I looked at the heatmap distributions of the Churn customers and the stay customers across the different categorical variables, some major issues stood out. The first, as demonstrated in the heat map image below, shows that there was a large proportion of Yes churn customers who were ‘No’ in these categories. This indicates that they were paying for internet service, but they were not utilizing all of the services that Telco offered. This further explains the concentration disparity discussed earlier.



The other issue that stood out from studying the heatmaps was the specter of collinearity between certain categories. This is demonstrated when looking at these two heatmaps. The ‘No internet service’ is exactly the same in both. This held true for the following features: 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', and 'StreamingMovies'. This was something I ended up exploring more in both the Capstone2_featureselection & Capstone2_LogReg_and_RFC_usingselectedfeatures notebooks.



In the heatmaps above, another interesting find is that 42% (1297 out of 3096) of the customers who had purchased fiber optic internet ended up leaving Telco. An issue that could be researched more would be what type of services the Month-to-month contract customers are purchasing. Of the month-to-month contract customers, 43% ended up leaving Telco. This can be seen in the heatmap below.



Models:

Baseline Model - Because this is a classification problem, I chose to use scikit-learn's LogisticRegression model as my baseline. I chose initially to use OneHotEncoder to transform the categorical data, because it fit well in a pipeline and, if the model were to be put into use, could handle new data with ease. I also chose to use StandardizedScaler to scale the numeric values. This initial model obtained a Yes recall score of .55. This model can be found in the 3.0_Capstone2_LogisticRegression_initialassessment notebook.

Correcting for Imbalanced Data Models - The information that I obtained from my baseline model indicated that it had not overfit. But I realized that I was dealing with an imbalance data issue. So to obtain a higher Yes recall score, I would need to address this issue. The next three models can be found in the 3.1_Capstone2_LogisticRegression_imbalanceddata notebook.

Still using a logistic regression model, I used the stratification process on the train test split to see if it produced a noticeable difference in the results of the model. Then, I implemented an oversampling technique, SMOTE. Finally, I used an undersampling technique, Nearmiss. The SMOTE and Nearmiss techniques both produced an increased Yes recall score (both were .79), but the SMOTE showed a slight drop in one f1-score while the Nearmiss showed a large reduction in both f1-scores.

RandomForest Models- I wanted to use an ensemble approach to this classification problem, so I decided to use scikit-learn's RandomForestClassifier. I first tried using the model with no changes to it. I continued to stratify my data on the train test split because of the imbalance issue. This model had a Yes recall of .51.

To be able to use the model's feature_importance, I was forced to stop using the OneHotEncoder. Instead, I switched to using get_dummies. This allowed me to see which features the random forest model was finding important.

Again, as with the logistic regression models, I trained random forest models using the SMOTE oversampling technique and the Nearmiss undersampling technique to see if they could help improve the Yes recall score. The RandomForest with SMOTE technique produced a Yes recall score of .52 while the RandomForest with Nearmiss technique produced a Yes recall score of .69.

Feature Selection Models- In the EDA notebook and the feature selection notebook, I identified certain features that might benefit a model by having them dropped or changed. In the Capstone2_LogReg_and_RFC_usingselectedfeatures notebook, I do just that. In this notebook, I used the baseline scikit-learn LogisticRegression model and the RandomForestClassifier. I used the same two models and the same methods to address the imbalance data (standard stratification, with SMOTE, and with Nearmiss) so I would be able to accurately compare the feature selection model with the models that I had trained and tested earlier with the complete Telco data set. The results were mixed, with Yes recall scores ranging from .49 to .79.

In this notebook, I also trained a 7th model using a LogisticRegression model with the SMOTE oversampling technique to see how the model would perform if I dropped the features that seemed to have collinearity. The model scored a Yes recall of .79.

Year 1 models- In my final model notebook, I trained two models: a scikit-learn LogisticRegression and a scikit-learn RandomForestClassifier. For the data for these models, I used just those customers who had been with Telco for 12 months or less. Using this data set solved the imbalanced data issue because the churn Yes to No ratio was 47% to 53%. It also allowed me to explore more thoroughly the reasons as to why such a large number of customers left Telco in the first year. The models scored .68 and .63 on the Yes recall.

Findings:

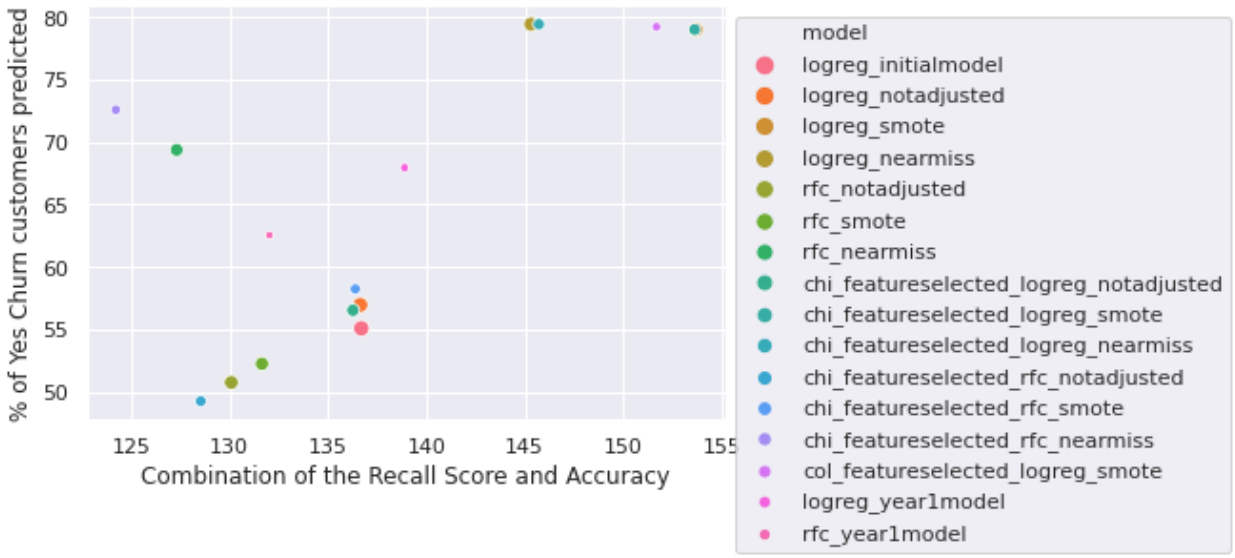
The Results- Below is a table that contains the results of all of the models I trained and tested for this project. There is also a scatterplot with all of the points plotted out for a visual representation of the different models and how well they scored on Yes recall and on total accuracy.

Note: All scores are based on the test data set.

Model	Class	Precision	Recall Score	f1-score
LogisticRegression w/SMOTE	No Churn	0.91	0.73	0.81
	Yes Churn	0.51	0.79	0.62
LogisticRegression w/SMOTE using features selected based on Chi2 tests	No Churn	0.91	0.73	0.81
	Yes Churn	0.51	0.79	0.62
LogisticRegression w/SMOTE using columns modified for collinearity	No Churn	0.90	0.70	0.79
	Yes Churn	0.49	0.79	0.60

LogisticRegression w/Nearmiss using features selected based on Chi2 tests	No Churn	0.89	0.62	0.73
	Yes Churn	0.43	0.79	0.56
LogisticRegression w/Nearmiss	No Churn	0.89	0.61	0.72
	Yes Churn	0.42	0.79	0.55
LogisticRegression using first year customer data only	No Churn	0.72	0.74	0.73
	Yes Churn	0.70	0.68	0.69
LogisticRegression baseline model	No Churn	0.85	0.91	0.88
	Yes Churn	0.70	0.55	0.62
LogisticRegression using just the stratification process on train test split	No Churn	0.85	0.88	0.86
	Yes Churn	0.63	0.57	0.60
RandomForestClassifier w/SMOTE using features selected based on Chi2 tests	No Churn	0.85	0.85	0.85
	Yes Churn	0.59	0.58	0.59
LogisticRegression using just the stratification process on train test split while using the features selected based on Chi2 tests	No Churn	0.85	0.88	0.86
	Yes Churn	0.63	0.57	0.60
RandomForestClassifier using first year customer data only	No Churn	0.69	0.76	0.72
	Yes Churn	0.70	0.63	0.66
RandomForestClassifier w/SMOTE	No Churn	0.84	0.89	0.86
	Yes Churn	0.64	0.52	0.57
RandomForestClassifier using just the stratification process on train test split	No Churn	0.83	0.90	0.86
	Yes Churn	0.64	0.51	0.57
RandomForestClassifier using just the stratification process on the train test split while using the features selected based on Chi2 tests	No Churn	0.83	0.90	0.86
	Yes Churn	0.64	0.49	0.56
RandomForestClassifier w/Nearmiss	No Churn	0.83	0.54	0.65
	Yes Churn	0.35	0.69	0.47

RandomForestClassifier w/Nearmiss while using the features selected based on Chi2 tests	No Churn	0.82	0.44	0.57
	Yes Churn	0.32	0.73	0.44



The 'Best'- There were five models that each scored .79 on the Yes recall which was the measure that I was focusing most on for this project. To further delineate which models would be considered best, I looked at the f1-scores for both the 'No' and 'Yes' classes. Using these metrics, the top 3 were all LogisticRegression models that utilized the SMOTE technique to address the imbalanced data issue. It is also worth noting that the top 50% of models were all LogisticRegression models, indicating that for this data set and this problem, overall the LogisticRegression model was more able to identify which customers would Churn.

Any of the top three models would serve Telco well if it went into production. In terms of simplicity of implementation, the LogisticRegression model with the SMOTE technique tested and trained on the original data would be easiest as it would require the least amount of pre-processing. Thus, I would consider it the best model, but any of the top three had a high enough Yes recall score with corresponding high f1-scores that it would benefit Telco in identifying customers who will leave.

Model Insights - For both the LogisticRegression model and the RandomForestClassifier model, a post-training analysis can be conducted on how the models used and weighted the features. This post-model analysis can sometimes yield insight for businesses. Below are three tables: two are of LogisticRegression models and the third is from the top-performing RandomForest model.

For the LogisticRegression models, I used coef_ to identify how the model was using different features. Each feature is given a value along with a positive or negative sign. A positive number indicates that the model viewed that feature as contributing to the probability that the customer would churn, while a negative number indicated that the model viewed that feature as contributing to the probability that the customer would not leave Telco.

LogisticRegression w/SMOTE using features selected based on Chi2 tests

Top Features Contributing to the 'Yes' Class	Weight	Top Features Contributing to the 'No" class	Weight
TotalCharges	0.86	Contract_Two year	1.56
MultipleLines_No phone service	0.85	tenure	1.52
InternetService_Fiber optic	0.41	Contract_One year	0.92
MonthlyCharges	0.40	TechSupport	0.64
PaperlessBilling_Yes	0.37	OnlineSecurity	0.60

These results were not overly shocking and reinforced the themes that had already been identified in the EDA. For example, for the features that were weighted towards the Yes class, 'TotalCharges' is at the top with an absolute value weight of 0.86. This might seem strange at first as there was a significant grouping of customers who stayed with Telco that had been with the company a significant amount of time and also paid a premium for their internet service and therefore paid a significant amount in 'TotalCharges'. But when this feature is considered in the context of the weighted features towards a No classification, it makes more sense. In the top features contributing to a No classification, the second-highest was 'tenure' with an absolute value weighting of 1.52. This is significantly higher than the 0.86 weighting, which means the model considered a person with a longer tenure with Telco more likely to fall into the No classification instead of the Yes classification. This means that the TotalCharges would only shift those customers towards the Yes who had a short tenure with Telco. This would take into account the significant number of customers who left Telco within a year of beginning service. That is why the chart below is interesting, because it looks specifically at the feature weight rankings for customers who had only been with Telco a year or less.

LogisticRegression using first year customer data only

Top Features Contributing to the 'Yes' Class	Weight	Top Features Contributing to the 'No" class	Weight
InternetService_Fiber optic	0.62	Contract_Two year	2.04
MultipleLines_Yes	0.40	Contract_One year	0.76
PaperlessBilling_Yes	0.38	tenure	0.75
MultipleLines_No phone service	0.37	TechSupport_Yes	0.42
StreamingTV_Yes	0.32	PhoneService_Yes	0.37

In this list above, 'TotalCharges' has been dropped from the highest weighted factor towards the Yes class. It does not even make the top 5 anymore. But fiber optic internet service is still present and has increased its weighting. No phone service and paperless billing have made it on both lists of the top features contributing to a Yes classification.

On the other hand, the top features which the model weighted as contributing to the probability that a customer would stay with Telco were almost the same for both lists. The difference is in their order and the size of the absolute value of their weighting. Customers' being signed up for a 2-year contract was weighted the most for both models. According to both models, the service that influenced the customers to stay with Telco the most was tech support.

For the RandomForestClassifier models, I used feature_selection_ to identify which features the model found most useful to create helpful splits for its decision trees. This list seems to echo the information from the lists produced by the LogisticRegression model coef_ .

RandomForestClassifier w/SMOTE using features selected based on Chi2 tests

Features in order of importance (top 5)
tenure
TotalCharges
MonthlyCharges
Contract_Two year
InternetService_Fiber Optic

Conclusion:

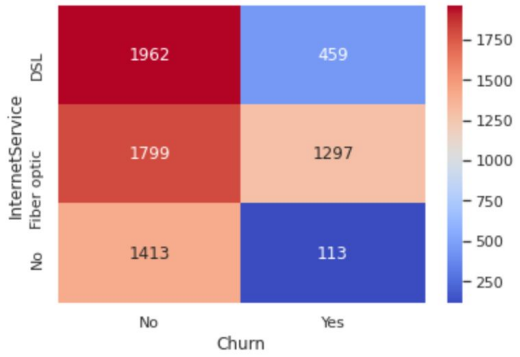
Future Work- I would have liked to have tried other models. I would have especially liked to have tried an unsupervised approach. Along the same line of thought, I would have liked to have tried to ensemble the LogisticRegression and RandomForest models with at least one more model.

I would have liked to have explored the high-end paying group of customers that had a long tenure with Telco. Along with this, I would have liked to have explored more the characteristics of the customers who left Telco within the first year. Both of these were outside the scope of this project.

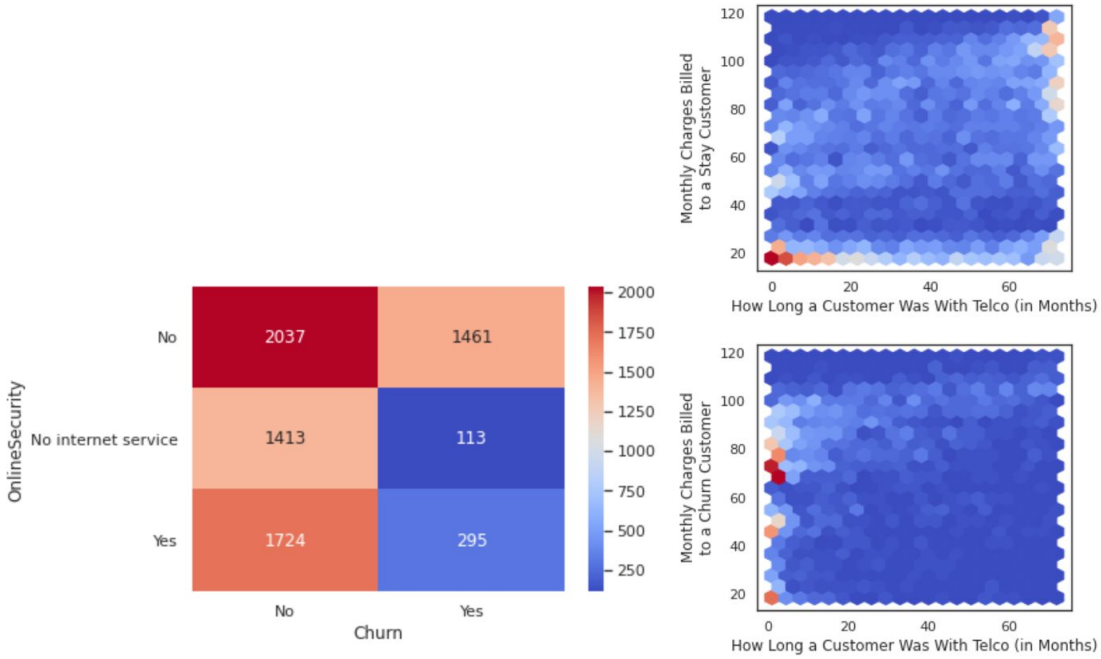
It would have been interesting to explore which Yes churn customers the models were having a tough time with. It would be interesting to compare the False negatives between the two models to see if they were having a difficult time with the same customers or if they were each able to identify different customers correctly and incorrectly.

Recommendations for the Client- As stated in the intro, if just 50% of customers that the best models identified as customers who will leave Telco ended up staying with Telco, then that would be a retention of $54,956.68/month, or close to $660,000 a year.

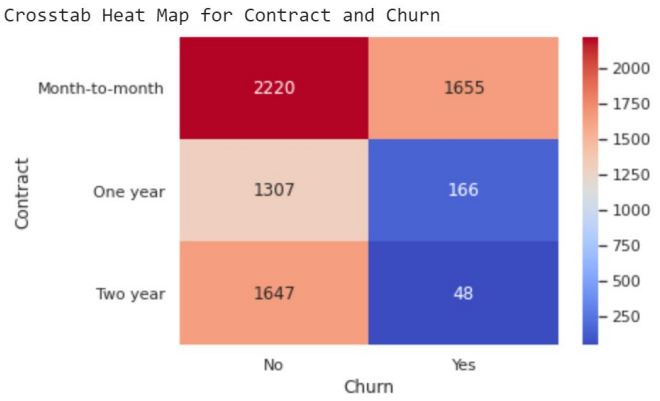
Per the information gleaned from the EDA as discussed on page 6 and reiterated with the heat map below, there needs to be further research done by Telco as to why 42% of their fiber optic customers are leaving. The post-model analysis, as discussed on pages 10-12, on the model coefficients (for the LogisiticRegression models) and the feature selection (for the RandomForestClassifier models) confirmed the need to review Telco's Fiber Optic service.



Why is it that there is a significant portion of the customers who purchase internet through Telco but do not sign up for Telco's higher-end services that end up leaving? This was elucidated on pages 4 & 5, and an example can be seen below in the heat map of a premium internet service (in this case Online Security) and the hex map that shows the density of the customers in the different price points in relation to how long they have been with Telco. The post-model analysis on pages 10-12 indicates that this is an issue as well. The LogisticRegression model placed a high absolute value weighting towards staying with Telco on customers who purchased Tech Support along with their internet service.



Telco should also look into eliminating the month-to-month payment plan. As highlighted on page 6 and illustrated with the heatmap below, according to the EDA, 43% of the customers who had a month-to-month payment plan ended up leaving Telco. The post-model analysis of the LogisticRegression models indicates that having customers sign up for a 1-year or 2-year contract can have a significant impact on customers staying with Telco over the long term. This can be seen on the charts on page 11.



Resources Used or Helpful:
This project was done using Google Colabs.

Packages- pandas, numpy, matplotlib, seaborn, scipy, sklearn, imblearn

Idea and base code for type and shape checking of train test split; the recommendation to look at the feature coef when using the LogisticRegression model; suggestion to use SMOTE and Nearmiss to address imbalanced data - A J Sanchez - Thankful for the private correspondence and the copious help!

Idea and base code for using the Chi-squared tests - Cornellius Yudha Wijaya
https://towardsdatascience.com/categorical-feature-selection-via-chi-square-fc558b09de43

Idea and base code for sklearn pipeline with OneHotEncoder and StandardScaler - Kevin Markham
https://youtu.be/irHhDMbw3xo

Base code for ROC curve and AUC score - Ajay Ramesh - found in answer to this stack exchange question: https://stackoverflow.com/questions/25009284/how-to-plot-roc-curve-in-python

Base code for the SMOTE and Nearmiss was from the imbalanced-learn documentation - https://imbalanced-learn.readthedocs.io/en/stable/auto_examples/index.html#model-selection

Idea and base code for RandomForest feature importances - Coder Unknown - found on Springboard's Data Science Career Track Jupyter Notebook: RandomForest_casestudy_covid19

Help with my Seaborn plots and the side by side plots came from - Jovian Lin - https://jovianlin.io/data-visualization-seaborn-part-2/