Machine Learning Engineer Nanodegree

Capstone Project

Christopher Meyer
July 14th, 2021

I. Definition

Project Overview

This project is an attempt to better understand customer churn in the telecommunications industry. Churn is simply another term for customer turnover, or defection. This report will do an analysis of customer churn using a publicly available dataset from Kaggle. The analysis will be conducted using machine learning models, tools, and metrics to gather insight from the data.

Problem Statement

The telecommunications industry is a highly competitive environment with a large number of companies providing bundles of various types of services to meet customer's needs. In today's environment, telco companies are not just providers of landline telephony services, but also providers of wireless telephony services as well as internet service, online security, cloud backup, entertainment services, device protection and more. The list of offered features continues to grow. Even though the telco company provides the customers with a variety of services, there are still a number of customers that churn. This is the problem we want to find answers for. What influences teleco customer churn?

This problem will be analyzed with 3 supervised machine learning models then each model will be evaluated for its performance. The performance of the models will be measured using various metrics. We will look at the which features in the dataset have the most influence to gather insights from the dataset. Finally, we will evaluate the top principal components of the dataset which have the greatest influence on churn and examine the underlying features in these components. With this information, we hope to have some evidence to support a targeted effort to use the insights to reduce customer churn.

Metrics

Accuracy and F-score will be used to determine how well the models are performing as compared to the baseline. Since the output of this problem is binary, (churn: yes or no) the accuracy and F-score will be an appropriate measurement since it takes into consideration the precision and recall scores which will be discussed in more detail later. Training time and predicting time will also be considered when comparing the models against each other. The metric results for each model will be reviewed and a decision will be made to determine which model performs the best overall. After, the best model is selected additional fine tuning to the model will be attempted to improve the performance of the model. Once the model is tuned, the metrics of the model will be compared with previous results to determine if an improvement was achieved. Finally for good measure, the Area Under the Curve (AUC) will be measured from the Receiver Operating Characteristic (ROC).

II. Analysis

Data Exploration

The dataset is in the form of a single .csv file and it comes from Kaggle located at https://www.kaggle.com/blastchar/telco-customer-churn. This dataset has a total of 7043 customers with 20 different features about these customers. Some of the features describe the type of customer such as gender, senior citizen, partner and dependents. The other features describe the different types of services that the customer subscribes to and the associated billing for these services. Although, the dataset is rather small considering a number of large telco companies have millions of customers, it should be sufficient to gain some insight from the data.

Below is a description of the features and labels.

Featureset Details

- **customerID**: 7590-VHVEG (example)
- **gender**: Male or Female
- **SeniorCitizen**: Whether the customer is a senior citizen or not (1, 0)
- **Partner**: Whether the customer has a partner or not (Yes, No)
- **Dependents**: Whether the customer has dependents or not (Yes, No)
- **tenure**: Number of months the customer has stayed with the company
- **PhoneService**: Whether the customer has a phone service or not (Yes, No)
- **MultipleLines**: Whether the customer has multiple lines or not (Yes, No, No phone service)
- **InternetService**: Customer's internet service provider (DSL, Fiber optic, No)
- **OnlineSecurity**: Whether the customer has online security or not (Yes, No, No internet service)

- **OnlineBackup**: Whether the customer has online backup or not (Yes, No, No internet service)
- **DeviceProtection**: Whether the customer has device protection or not (Yes, No, No internet service)
- **TechSupport**: Whether the customer has tech support or not (Yes, No, No internet service)
- **StreamingTV**: Whether the customer has streaming TV or not (Yes, No, No internet service)
- **StreamingMovies**: Whether the customer has streaming movies or not (Yes, No, No internet service)
- **Contract**: The contract term of the customer (Month-to-month, One year, Two year)
- **PaperlessBilling**: Whether the customer has paperless billing or not (Yes, No)
- **PaymentMethod**: The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
- **MonthlyCharges**: The amount charged to the customer monthly
- **TotalCharges**: The total amount charged to the customer

Labelset Details

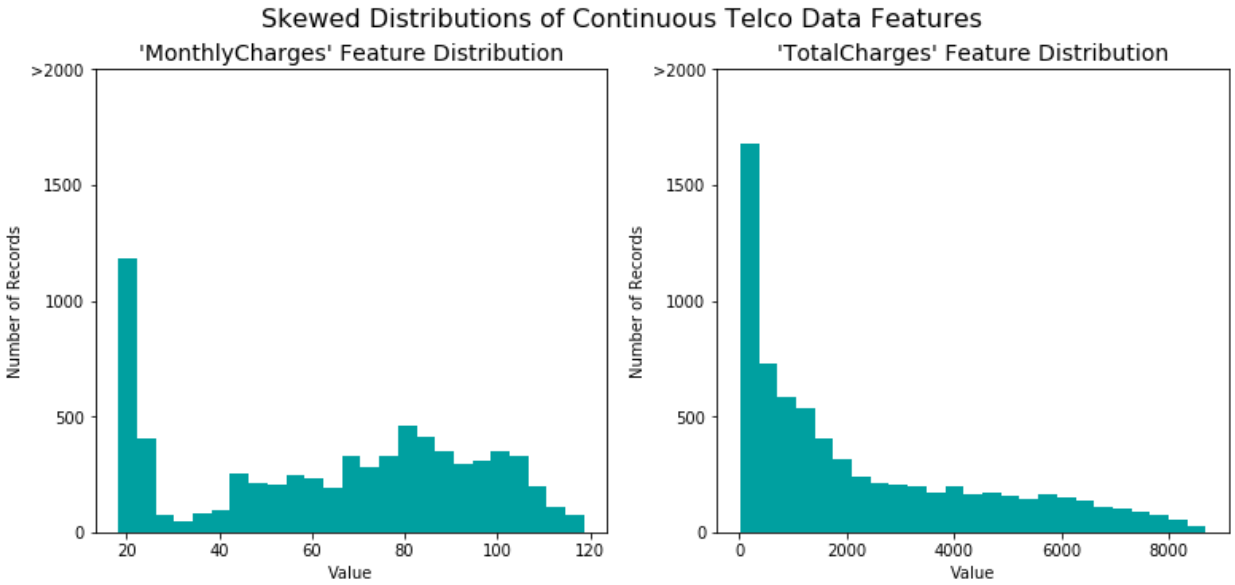
- **Churn**: Whether the customer churned or not (Yes or No)

After exploring the dataset, 11 customers were found to have missing data. The `TotalCharges` data was stored as text rather than numerical data. Much of the dataset had categorical values. All of these issues were addressed in the Data Preprocessing section of this document.

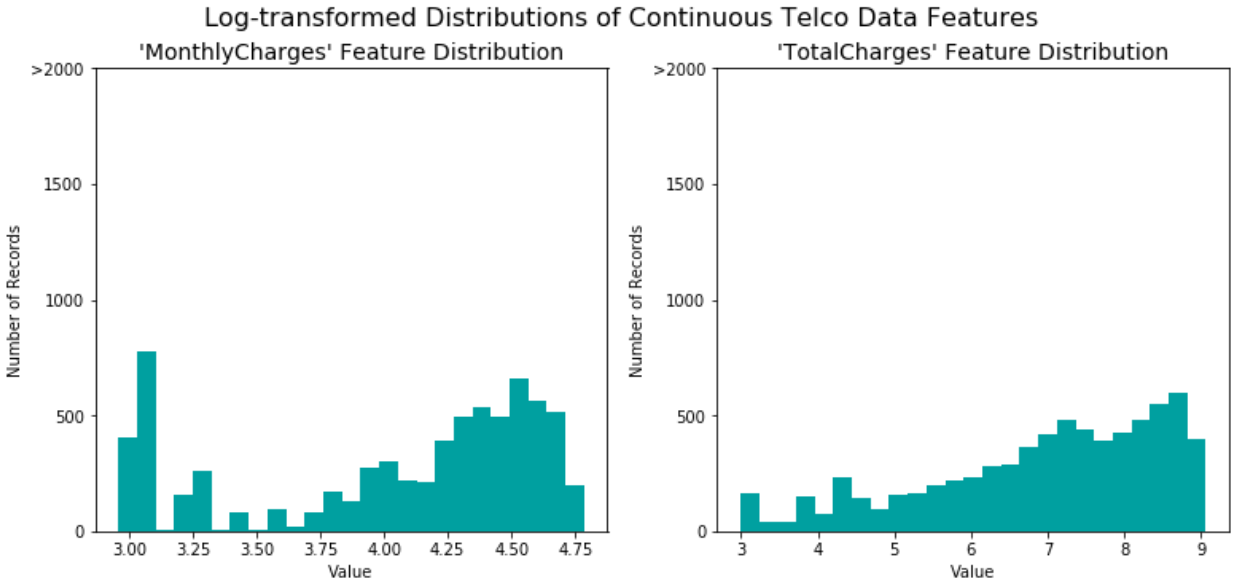
Exploratory Visualization

To determine if there were any abnormalities in the continuous data, a histogram of the data was plotted. The graphical representation of `TotalCharges`, and `MonthlyCharges` revealed that the data was skewed.

For highly-skewed feature distributions such as `TotalCharges`, and `MonthlyCharges`, it is common practice to apply a logarithmic transformation on the data so that the very large and very small values do not negatively affect the performance of a learning algorithm. Using a logarithmic transformation significantly reduces the range of values caused by outliers. Care must be taken when applying this transformation because the logarithm of 0 is undefined, so we must translate the values by a small amount above 0 to apply the logarithm successfully.



After applying the logarithmic transformation, the data appeared to be relatively normalized.



Algorithms and Techniques

For this problem, 3 supervised learning models were used to determine which one would provide the best results. Below is a description of the strengths and weaknesses of each model and the reason for selecting the model.

Random Forest

- The strength of Random Forest: it is very intuitive and simple to use. It provides interpretable predictions. It performs well on a large number of labeled features. It calculates results quickly and handles unbalanced and missing data well.
- The weakness of Random Forest is that it tends to overfit on noisy data.
- Random Forest was selected as a candidate for this problem because of the large number of features and its ability to provide insightful results that can be useful to target potential churn.

Logistic Regression

- The strength of Logistic Regression is the ability to predict binary outcomes such as yes or no (1 or 0). It can work on both linear and non-linear relationships between the input and output variables.
- The weakness of Logistic Regression is that performance decreases if the model has too many non-linear relationships. It does not adapt well to more complex situations. It must be tuned to avoid overfitting.
- Logistic Regression was chosen as a good candidate for this problem since it does well when predicting a binary outcome (churn or no churn).

XGBoost

- The strength of XGBoost is its ability to make predictions fast and trains rather quickly. It is capable of handling sparse data whereas many other models do not handle this well. It can also be used across multiple processors to increase efficiency and reduce training time. It can handle structured and unstructured data.
- The weakness of XGBoost is the fact that it can be difficult to determine the logic behind a prediction because of the use of several weak learners that are combined to produce a prediction. It also does not have deep analysis such that it truly understands the system of patterns that predict a result.
- XGBoost was chosen as a candidate for this problem because it works well with classification type problems.

Benchmark

To intelligently evaluate each model's ability to predict churn, a benchmark was established using accuracy and F-score on the dataset. Clearly the goal is to accurately predict churn which is the True Positive case. Intuitively, it is easy to believe the higher the accuracy the better the

results, however it is necessary to consider all possible results by evaluating a confusion matrix of possibilities. As shown below:

	Predicted Churn: YES	Predicted Churn: No
Actual Churn: YES	True Positive	False Negative
Actual Churn: NO	False Positive	True Negative

If the model falsely predicts a customer is about to churn (False Positive) and they do NOT it is not a huge loss to the telco company since the customer intends to keep their services but is perhaps offered a sweeter deal. If the model predicts that the customer will NOT churn and indeed, they stay with the telco company (True Negative) there is no loss or gain here. However, if the model predicts the customer will NOT churn, but they DO churn (False Negative) then this is the biggest loss to the company because the company loses all revenue from that customer. False Negatives are the outcomes that need to be avoided the most!

If the model produced a moderately good prediction accuracy of 70% it might be considered acceptable. This would mean that the model correctly predicts the customers who will churn and NOT churn with 70% accuracy. However, if the other 30% of predictions fall into the False Negative category, then the telco company will lose over 2,100 customers, in this case because of false predictions. Because this outcome can be detrimental to the company, the model needs to be sensitive to False Negatives. This can be done with a metric known as Recall, but measuring Recall on its own can have its own problems. The F-score which takes into account Precision and Recall scores is probably the best measurement in this situation. To further understand this, some terms need to be defined.

Accuracy measures how often the classifier makes the correct prediction. It's the ratio of the number of correct predictions to the total number of predictions (the number of test data points).

Accuracy = (True Positives + True Negatives)/(Total number of records)

Precision is a ratio of true positives (customers predicted to churn and actually do churn) to all positive predictions (all customers that are predicted to churn regardless if the classifier gets it right or wrong).

Precision = True Positives/(True Positives + False Positives)

Recall (AKA sensitivity) is a ratio of true positives (customers predicted to churn and actually do churn) to all the customers that actually did churn and actually did not churn, in other words it is the ratio of:

Recall = True Positives/(True Positives + False Negatives)

Taking into consideration the various metrics described above, the F1 score uses the harmonic mean (weighted average) of the Precision and Recall scores to produce a number between 0 and 1, with 1 being the highest F-score.

F-score = $(1 + \beta^2) * (precision * recall)/((\beta^2) * precision + recall)$
where $\beta = 0.5$

As a benchmark for the machine learning models, the accuracy and F-score was used.

III. Methodology

Data Preprocessing

Since only 11 customers were missing data in this dataset, these customer records were removed. The `TotalCharges` data was stored as an object so a correction was made to change it to a float.

Many of the features in this dataset were categorical in nature, one-hot encoding was performed to convert the data into numerical format. Typically, machine learning models only do well with numerical information, so the best way to convert these non-numerical values into numbers is to use a one-hot encoding scheme. One-hot encoding creates a "dummy" variable for each possible category of each non-numeric feature. For example, assume we have a customer-feature that could have a value of "A", "B", or "C". We then encode this feature into 3 separate features customer-featureA, customer-featureB, and customer-featureC.

customer-feature ---> A B C <----customer-feature is broken into 3 separate features
A	--->	1	0	0
B	--->	0	1	0
C	--->	0	0	1

Some of the numerical data was continuous, as a result a normalization method was applied using MinMaxScaler. It is often good practice to perform some type of scaling on numerical features. Applying a scaling to the data does not change the shape of each feature's distribution; however, normalization ensures that each feature is treated equally when applying supervised learners. It is important to note that once scaling is applied, observing the data in its raw form will no longer have the same original meaning.

In order to create training and testing sets for the models to learn, the data was split randomly. 80% was used for training and 20% was used for testing.

Implementation

A training and predicting pipeline were implemented to feed all 3 models the training and testing data. Each model was provided a sample size of the dataset in the proportions of 1%, 10%, and 100% of the dataset. These dataset sizes were used to determine the time to compute the final results for each model using three sample sizes for comparison. The output of the pipeline provided the measured results of the predictions provided by each model and the time to compute. A custom package *visuals.py* was imported to provide graphical results of the data and metrics results for each model. The visuals provided by this package included histograms of the continues data and bar graphs of the metrics for each model (see Model Evaluation and Validation).

The metrics used for measuring the output of these models are Accuracy, F-score, Precision, and Recall. After selecting the best model based on the metrics of each model, the area under the curve (AUC) was calculated from the Receiver Operating Characteristic (ROC) as a final measurement of performance.

In addition, a correlation graph of all categorical features was used to gather insight on the features that had the most influence on churn. These features were also compared to the feature importance for each of the 3 models used.

Lastly, Principal Component Analysis (PCA) was used to examine the number of features that explained 80% of the variance in the dataset. An examination of the top 3 principal components was done to see the makeup of each principal component to determine the highest weighted features that had the most influence on the component.

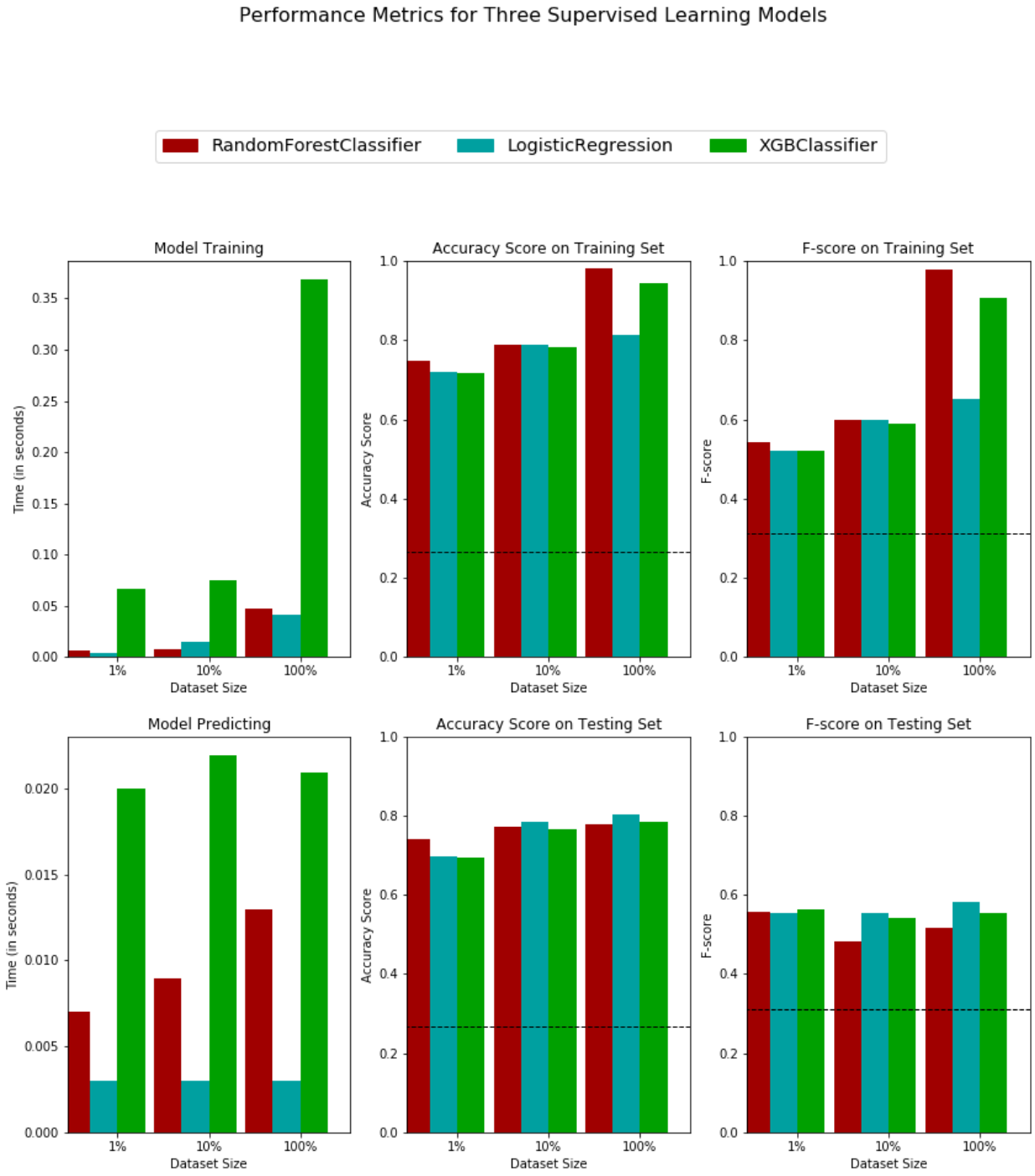
Refinement

After selecting the best model from the 3 models implemented in the pipeline, a grid search tool was used to run through an exhaustive analysis of all parameters in an effort to tune the model for better results. Unfortunately, after the model was optimized for the best results, it did not improve over the unoptimized model.

IV. Results

Model Evaluation and Validation

To better evaluate the 3 models used, a graphical representation of the metrics was created.



For this problem Logistic Regression was chosen as the winner. It was significantly faster at training than XGBoost. It also made predictions faster than Random Forest and XGBoost. Logistic Regression scored the highest on accuracy and F1 score when predicting results on 100% of the test data. Clearly none of the models took a very long time to train or test, but if there was a bigger dataset from a large telco business then running these classifiers on millions of customer records could take a lot longer.

After selecting the winner, the model was tuned for improvement however the results did not improve as shown below:

```
Unoptimized model
------
Accuracy score on testing data: 0.8038
F-score on testing data: 0.6287
Classification Report
               precision    recall  f1-score   support

           0       0.84      0.91      0.87      1036
           1       0.66      0.52      0.58       371

   micro avg       0.80      0.80      0.80      1407
   macro avg       0.75      0.71      0.73      1407
weighted avg       0.79      0.80      0.80      1407


Optimized Model
------
Final accuracy score on the testing data: 0.8038
Final F-score on the testing data: 0.6287
Classification Report
               precision    recall  f1-score   support

           0       0.84      0.91      0.87      1036
           1       0.66      0.52      0.58       371

   micro avg       0.80      0.80      0.80      1407
   macro avg       0.75      0.71      0.73      1407
weighted avg       0.79      0.80      0.80      1407
```

As a sanity check, the area under the curve (AUC) was calculated from the Receiver Operating Characteristic (ROC) as a final measurement. If the area under the curve (AUC) is 0.5 then the model failed to distinguish between customers that will churn vs no churn. In other words, it makes 50% right and 50% wrong predictions which is not helpful.

Below is a general rule for AUC ratings[1]:

- AUC: 0.5 (failed)
- AUC: 0.7 – 0.8 (good)
- AUC: 0.8 – 0.9 (excellent)
- AUC: 0.9+ (outstanding)

The calculated AUC for the Logistic Regression model was 0.713. This is a fairly good result, but my expectation is that it would improve a bit more if there was a bigger dataset to work with.

Justification

Considering the benchmark accuracy score: 0.2658 and F-score: 0.3115, there was a significant improvement when comparing results to the Logistic Regression model which had an accuracy score: 0.8038 and F-score: 0.6287.

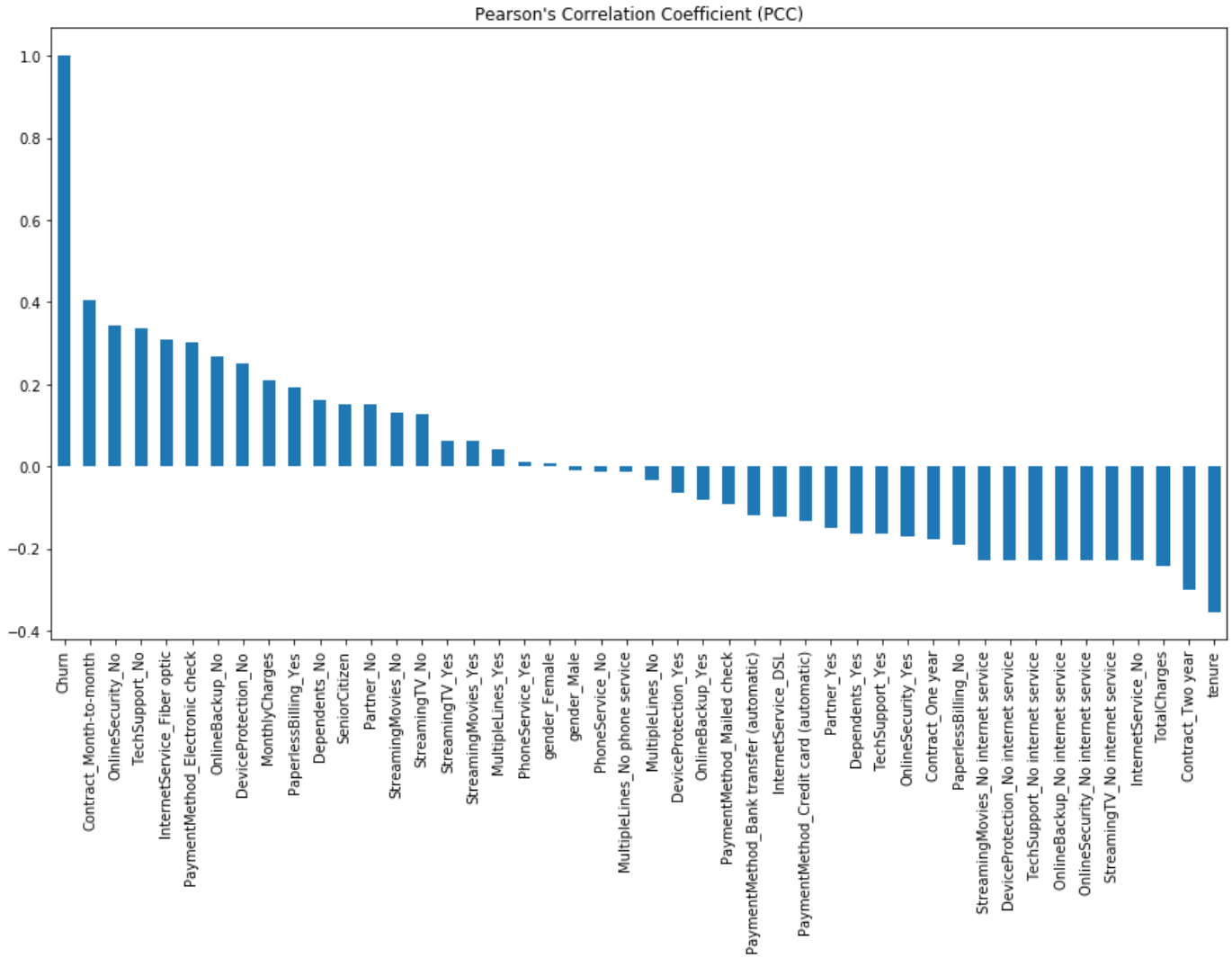
The outcome of the metrics show that this is an effective method to predict churn. With the insights that are gathered from exploring the additional data such as the correlation of all categorical features, feature importance, and principal components, the telco can make a targeted effort using this data to approach its customers with incentives to reduce churn. After making deliberate efforts to reduce churn based on the data insights, a further review of churn should be conducted to see if the targeted efforts made an impact on improving churn.

V. Conclusion

Free-Form Visualization

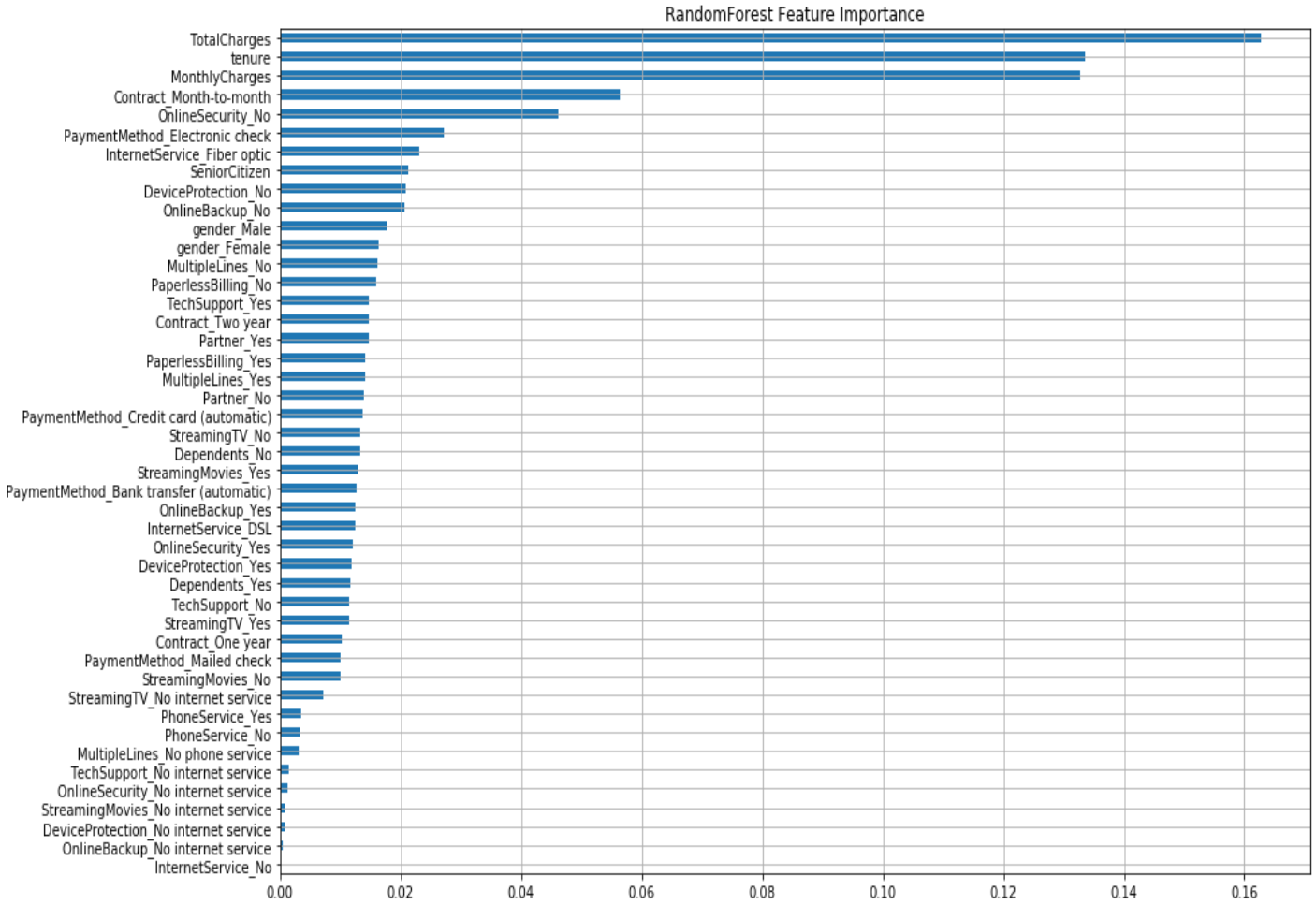
When examining the underlying data in more detail, additional insights become more apparent. In the following correlation graph, Pearson's correlation coefficient is calculated which is the covariance of two features (Churn and another feature) divided by the product of their standard deviations which results in a number between -1 and 1. We find that `tenure`, `Contract_Two year`, and `TotalCharges` help reduce churn. On the other hand, `Contract_Month-to-month` increases churn which is an expected outcome. We also see `OnlineSecurity_No` and `TechSupport_No` may increase churn.

[1] https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/#:~:text=1%20An%20AUROC%20of%200.5%20%28area%20under%20the,the%20figure%20above%29%20corresponds%20to%20a%20perfect%20classifier


Pearson's Correlation Coefficient (PCC)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
Churn
Contract_Month-to-month
OnlineSecurity_No
TechSupport_No
InternetService_Fiber optic
PaymentMethod_Electronic check
OnlineBackup_No
DeviceProtection_No
MonthlyCharges
PaperlessBilling_Yes
Dependents_No
SeniorCitizen
Partner_No
StreamingMovies_No
StreamingTV_No
StreamingTV_Yes
StreamingMovies_Yes
MultipleLines_Yes
PhoneService_Yes
gender_Female
gender_Male
PhoneService_No
MultipleLines_No phone service
MultipleLines_No
DeviceProtection_Yes
OnlineBackup_Yes
PaymentMethod_Mailed check
PaymentMethod_Bank transfer (automatic)
InternetService_DSL
PaymentMethod_Credit card (automatic)
Partner_Yes
Dependents_Yes
TechSupport_Yes
OnlineSecurity_Yes
Contract_One year
PaperlessBilling_No
StreamingMovies_No internet service
DeviceProtection_No internet service
TechSupport_No internet service
OnlineBackup_No internet service
OnlineSecurity_No internet service
StreamingTV_No internet service
InternetService_No
TotalCharges
Contract_Two year
tenure

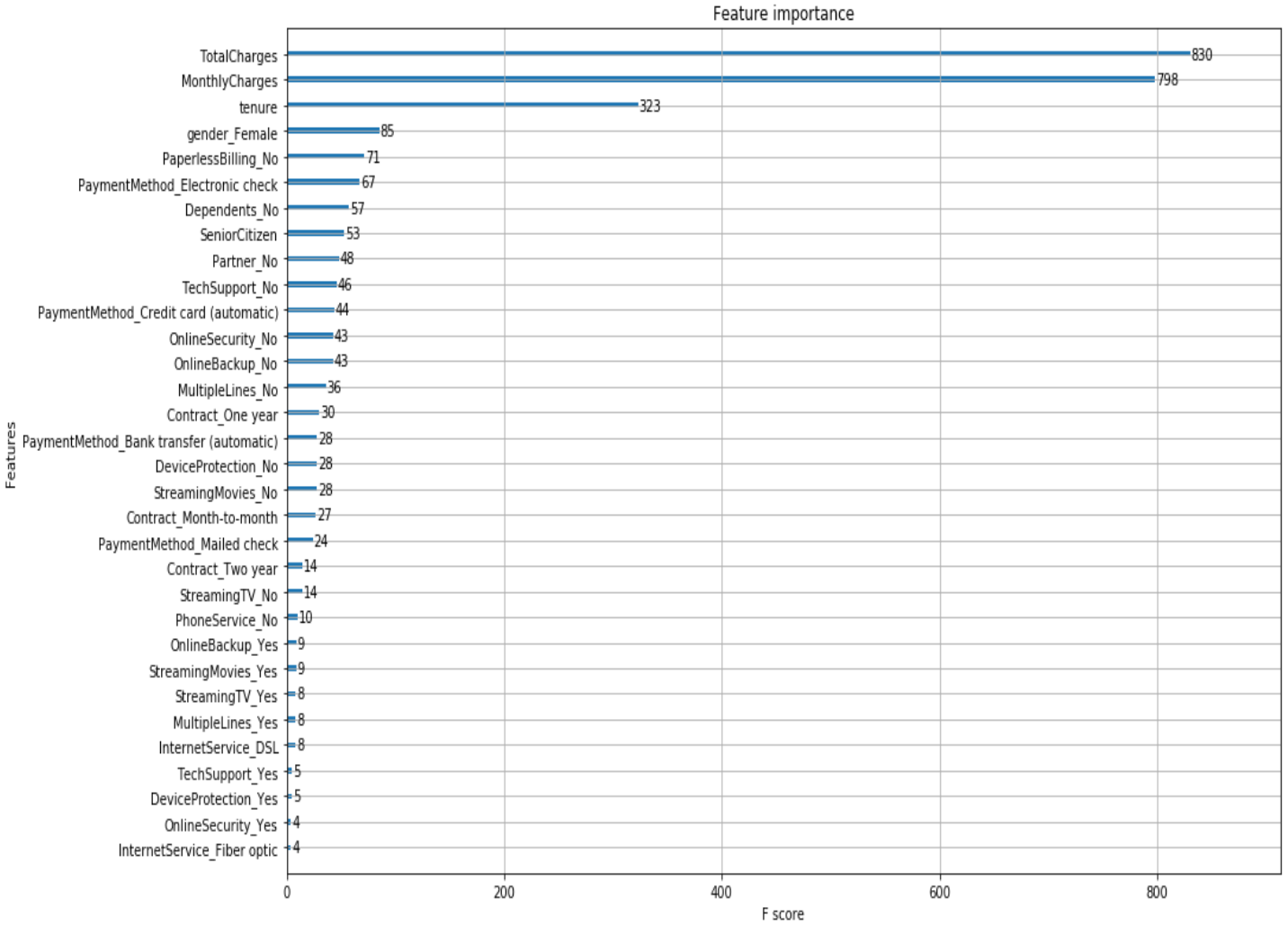
Below is a deeper dive into the data by looking at the feature importance of each model. Feature importance is basically the amount of influence that a feature has on the prediction of churn. The higher the weight the more important that feature is when making predictions.

Random Forest Feature Importance:



From the RandomForest model, we see that `TotalCharges`, `tenure`, `MonthlyCharges`, and `Contract_Month-to-month` have the most influence on churn with respect to all other features. We also see that `OnlineSecurity_No` makes into the top 5 most influential features just like the PCC graph shown before.

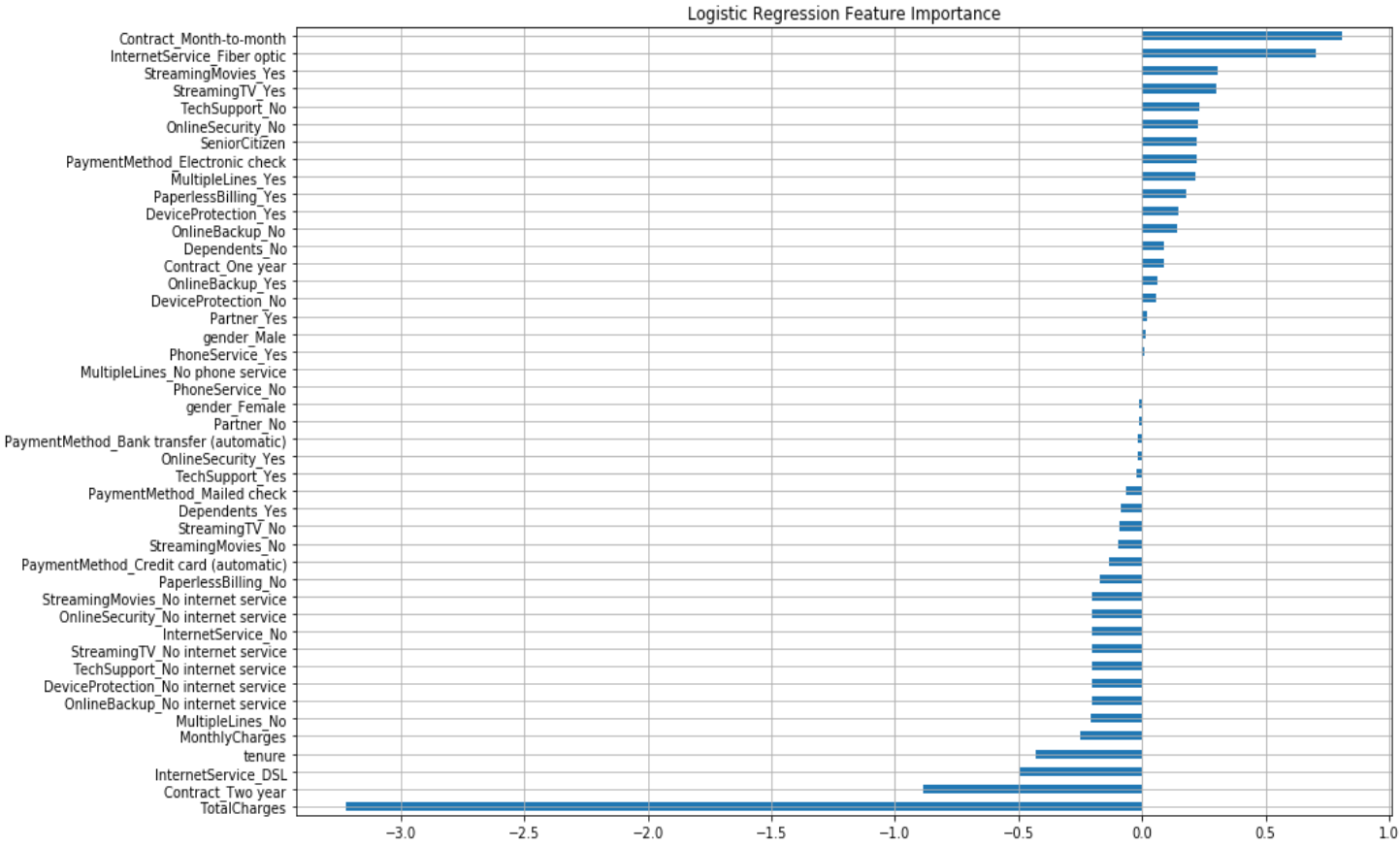
XGBoost Feature Importance:



From the XGBoost model, we see that this model largely agrees with the RandomForest model when comparing the top 3 most influential features that affect churn. It appears that it is only these 3 that make the most notable impact.

UDACITY
Christopher Meyer
7/14/2021

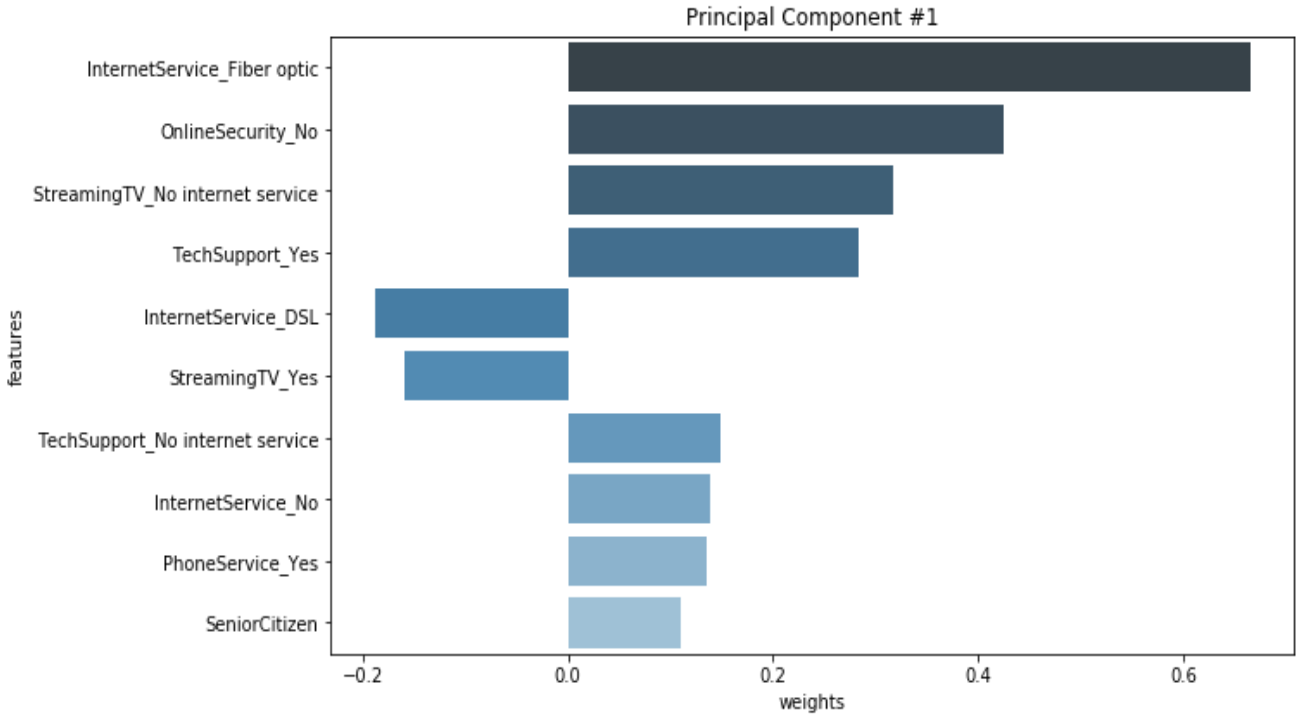
Logistic Regression Feature Importance:

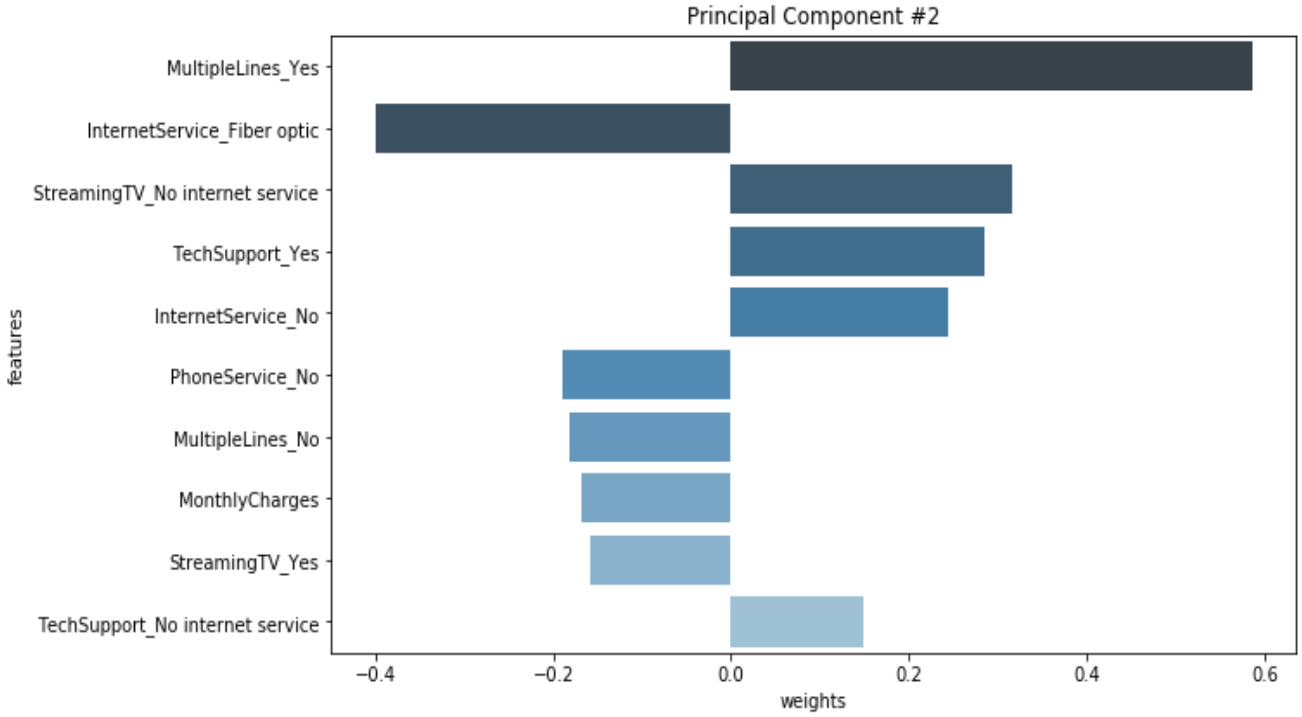


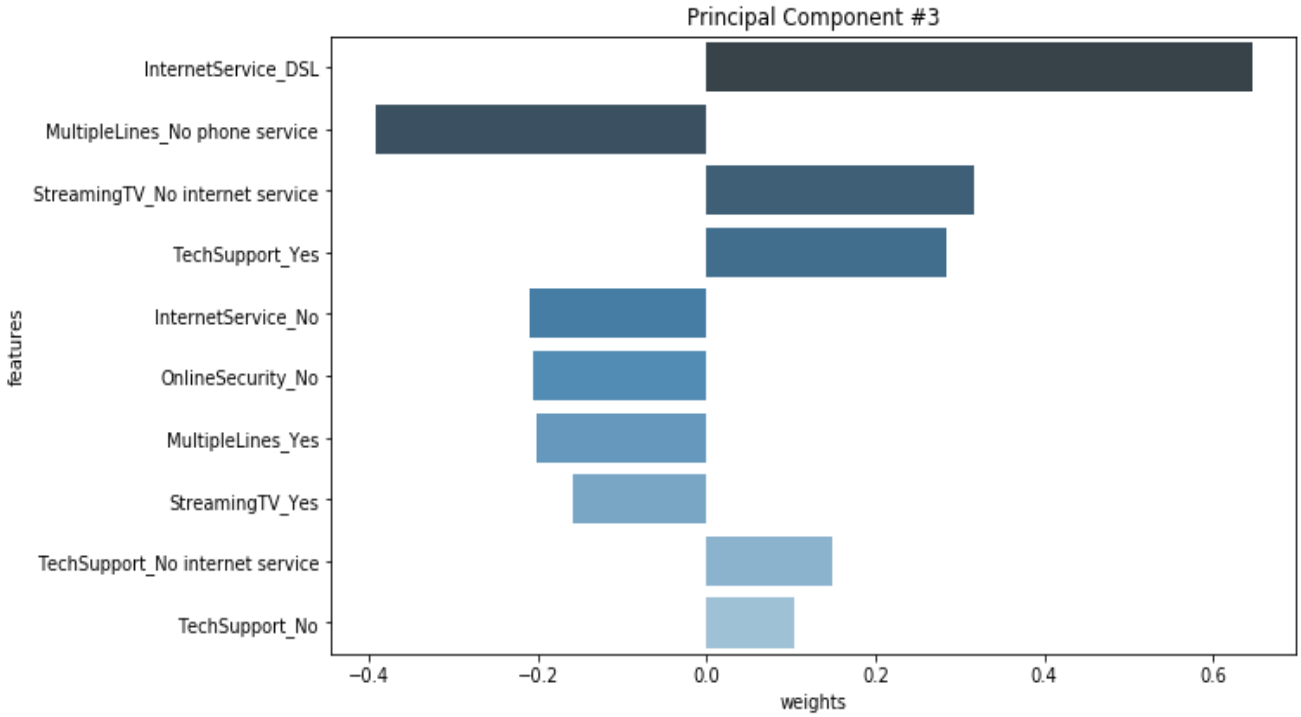
From the Logistic Regression model, we see that `Contract_Month-to-month` and `InternetService_Fiber optic` increases churn whereas, `TotalCharges` and `Contract_Two year` largely reduces churn. We should note that `TotalCharges`, `tenure` and `MonthlyCharges` are in the top 5 influencers in reducing churn which complements what we found in the other models. Intuitively the features that reduce churn and increase churn make sense. What we discover is that `InternetService_DSL` is a surprise factor in reducing churn and `InternetService_Fiber optic` is a surprise factor in increasing churn! These two features are closely related in that they are types of internet service, but it is interesting how each could have a polar opposite effect on churn. Fiber optic service would be extremely fast and DSL might be slow to moderate speed. Perhaps the cost of fiber optic service is much more than DSL. Perhaps the reliability of one service is far better than the other. The data by itself does not give us the answers. This would require some further investigation by the sales/marketing team and IT department.

Next, we will look at Principal Component Analysis (PCA) which is a method to reduce the number of features by creating "principal components". The method attempts to find many similar or redundant features and combines them to create a new feature set which has fewer features than the original dataset.

This method uses a linear transformation of features which is projected onto a reduced PCA space and captures the largest variances in the data without discarding any data. Below are the top 3 principal components and a visual breakdown of the features that comprise these components.







Once again, we see that `InternetService_DSL` and `InternetService_Fiber optic` standout as big influencers in all 3 PCs. There may be some further correlation with `StreaminTV_No internet service` and `TechSupport_Yes`.

Reflection

The following steps summarize the processes used for this project:

1. Visualize the data for anomalies
2. Clean and preprocess the data
3. Implement a training and testing pipeline
4. Tune model
5. Analyze results
6. Gather insights

The insights gathered from step 6, were the most interesting parts of this project. I particularly liked the visuals from the Correlation Graph, Feature Importance, and Principal Component Analysis. It was also very challenging to understand why some of the features had more influence on churn than others. At the same time there were some features that were quite easy to understand how they influenced churn.

As an engineer in the telco business, I do not pretend to know the marketing and influence of particular products on customers. I think that with the insights I've gathered it would be an interesting conversation with those who are involved with market and sales to get their input

on these feature correlations and feature importances. With the data that I can present to them, I think a targeted effort to decrease churn can be of great importance going forward. In fact, it would be interesting to see if we could intercept customers before they churn and offer them a service that we see helps prevent churn.

Improvement

As I look back, I believe that if I was able to do this kind of analysis on a much larger dataset of a million customers or more, then there would be a significant improvement in the prediction accuracy and F-score.

In an effort to improve the metrics, I would also do some additional tuning to the other models to see if they could make some improvements over Logistic Regression.

Finally, I would consider doing some additional analysis on `TotalCharges`, `tenure`, and `MonthlyCharges`. Since we have learned that these features have a large influence on churn, it would be very useful to see if there are any clear thresholds within each feature that indicate when a customer might churn. For example, at what point in a customer's tenure do they most often churn? Is there a monthly charge or total charge that crosses a threshold and influences a customer to churn? These are some additional avenues of analysis that could be done to further understand churn.