Udacity Machine Learning Nanodegree 2020 Capstone Report

Customer Retention in Telecommunications

Michael George

March 2020

I. Definition

Project Overview

Customer churn occurs when a customer (player, subscriber, user, etc.) ceases his or her relationship with a company.

The ability to predict that a customer is at a high risk of churning, while there is still time to do something about it, represents a huge additional potential revenue source for every business.

The full cost of a churning customer includes both lost revenue, marketing costs involved with replacing those customers with new ones and that the costs of initially acquiring that customer may not have already been covered by the customer's spending to date. It is always more difficult and expensive to acquire a new customer than it is to retain a current paying customer.

Businesses that fail to address churn suffer further debilitating consequences in reduced attractiveness to investors and doubts about their future viability. It's essential to measure, monitor, and reduce churn. Reducing churn is a key business goal of every business.

In order to succeed at retaining customers, the business must be able to
(a) predict in advance which customers will churn; and
(b) know which actions will have the greatest retention on each particular customer.

So the main objectives of this project will be to use Machine Learning techniques to classify 'churning' customers and their attributes for a given business dataset.

Problem Statement

Every business has customers that cease doing business with them. Failing to deal with churning customers has major consequences on a business, so the ability to predict and inhibit these customers from leaving is a must.

The business goal of this exercise is to apply Machine Learning techniques to:

1. Analyze customer specific data to understand who could be the next potential customers to leave the business.
2. Find what attributes contribute to the higher churn rate of customers and what could be some of the solutions to address this.

Metrics

The evaluation metric for this problem will be accuracy score.

When tuning the model we will factor in precision and recall to limit the number of false negatives and positives which lead to misclassification of churning customers.

II. Analysis

Data Exploration

A dynamic data sample can be found on the Kaggle website:
https://www.kaggle.com/blastchar/telco-customer-churn

After loading the csv file and doing a quick scan there are some columns with values and some with data.

A quick look at the dataframe reveals it contains a total of 7043 rows. Each row is unique for a customer and is identified using customerID. The dataset contains a total of 21 columns:

- CustomerID - Customer ID
- Gender - Customer gender (female, male)
- 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
- Churn - Whether the customer churned or not (Yes or No)

The target column for classification is ‘Churn’.

A further breakdown of features reveals this dataset has 18 categorical features; 6 binary (yes/no), 2 with 2 unique values, 9 with 3 unique values and one with 4 unique values.

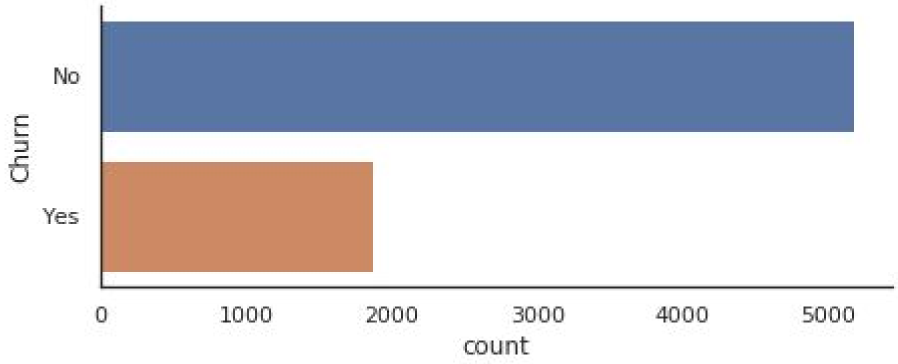
The next step is to clean up the data. The df.info() reveals no missing data but the TotalCharges column is an object and should be numeric. So after converting the column to numeric we find 19 values are null which is a low enough value to either drop the rows entirely or impute the mean. I choose to impute the mean.

The senior citizen data is numeric not categorical (with 1 referring to being a senior). I will convert it to categorical data if required later.

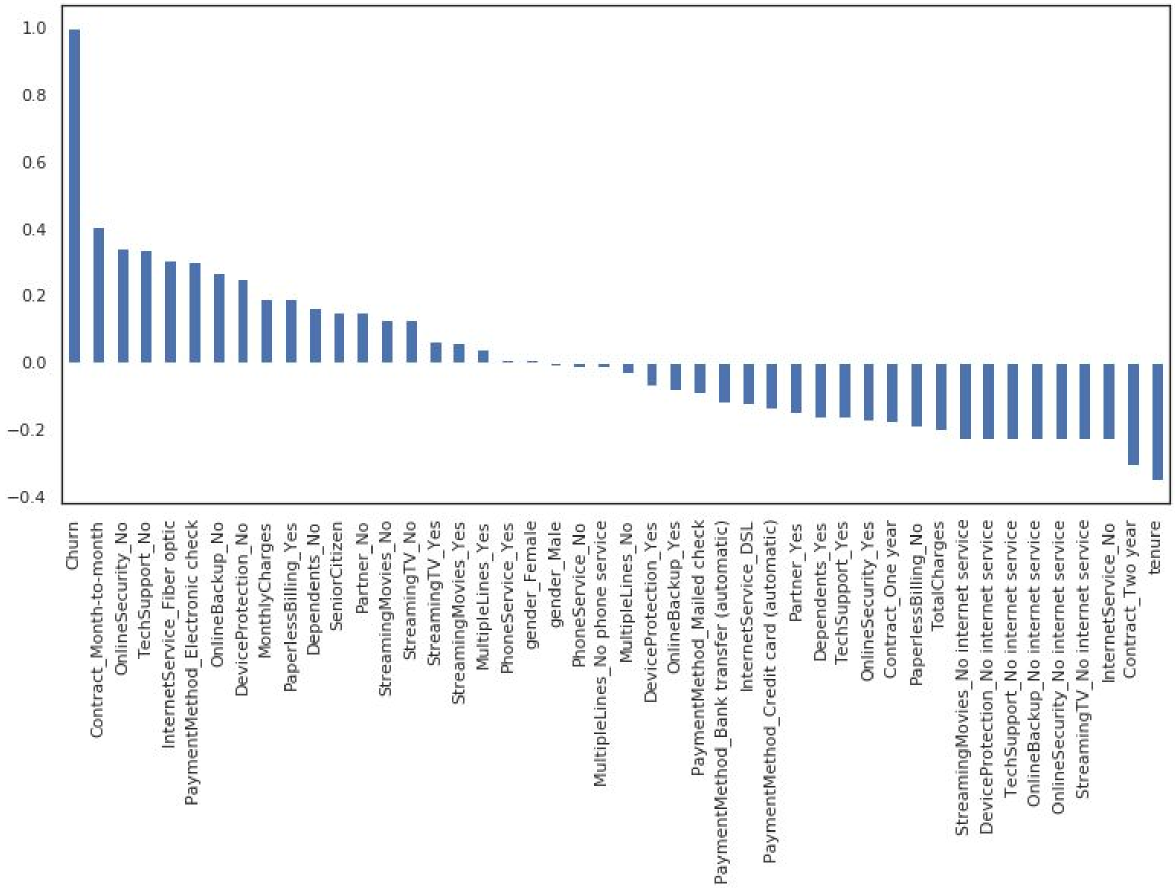
Exploratory Visualization

First we plot the target column, Churn, to find out how many customers have churned across the whole dataset.

We find that 26.5% (1869) of our 7043 customers have churned over the lifetime of the dataset.



Next we’ll look at the correlation of variables with the Churn feature by converting each categorical variable to indicator variables.



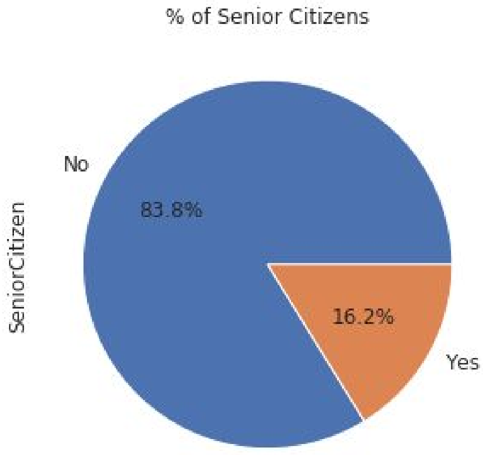
Here we can see that month to month contracts, absence of online security and tech support seem to be positively correlated with churn. While, tenure, two year and the availability of internet service contracts seem to be negatively correlated with churn.

We'll now break down some of the categorical features, starting with the types of customers.

From the correlation chart above dependants and senior citizens are the higher correlated demographics. Let's start with senior citizens.

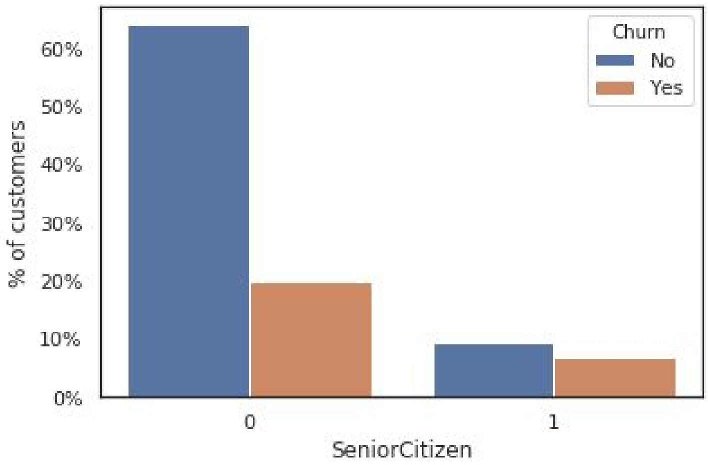
Senior Citizens

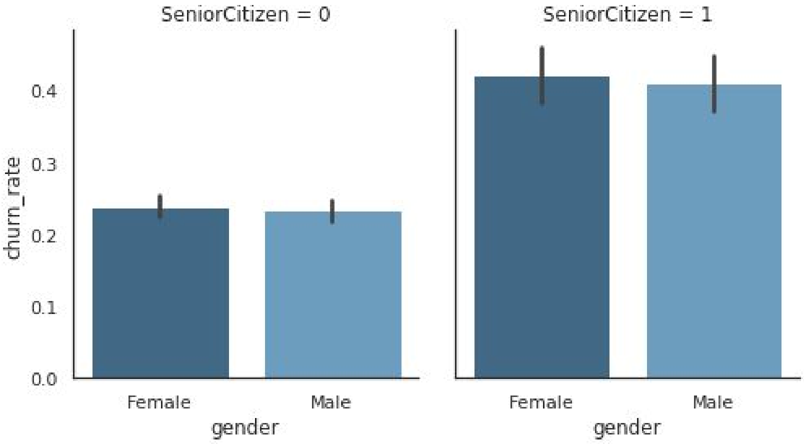
As a total of the population there are only 16% of the customers who are senior citizens. Of the population and of those 16% which are likely to churn?



The ratio of non-seniors churning to not churning is far lower than the ratio of seniors. As a proportion of the senior population it's a 60:40 split on whether they will churn or not. This will require further investigation.

We also have gender data so let's break the populations down by gender to see if there is any weightings either way.

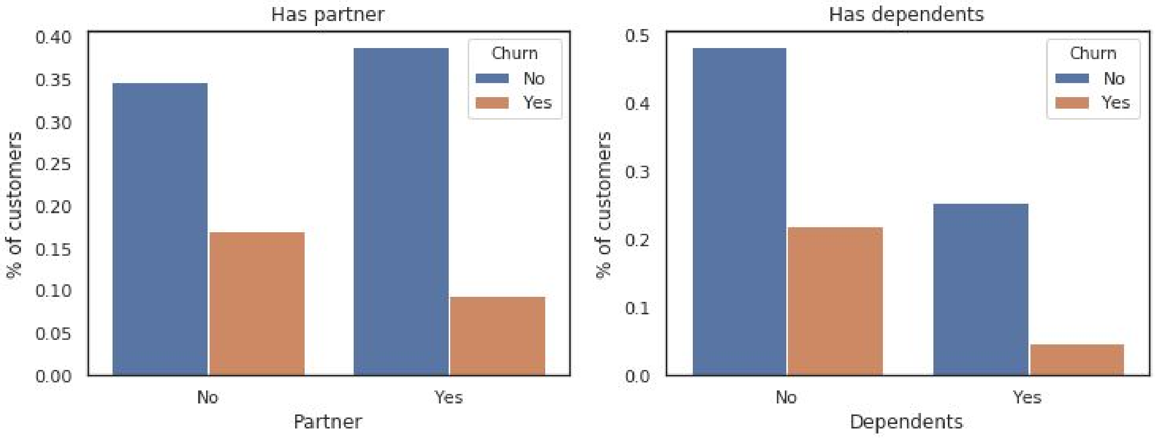




Gender is equally weighted across the population, so is not indicative of churn. The next category to look at is partners and dependants.

Partners and Dependants

Let's plot some bar graphs of churning and non-churning customers for both whether they have a partner and whether they have dependents and see how it affects churn.

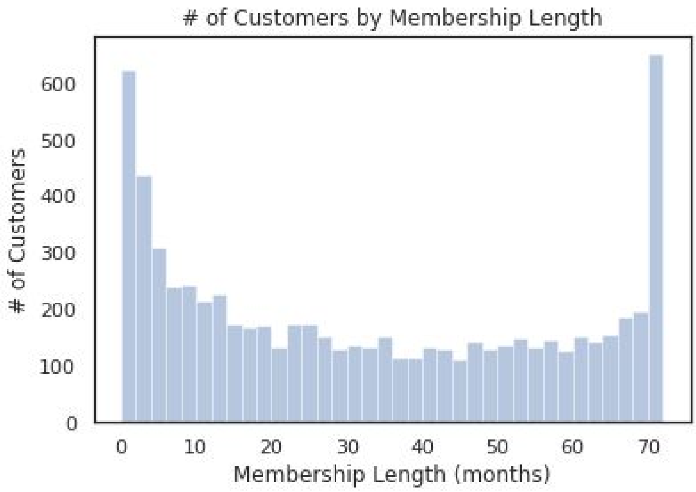


From the Partner graph we can see that customers that don't have partners are more likely to churn. From the dependents graph, customers without dependents are more likely to churn.

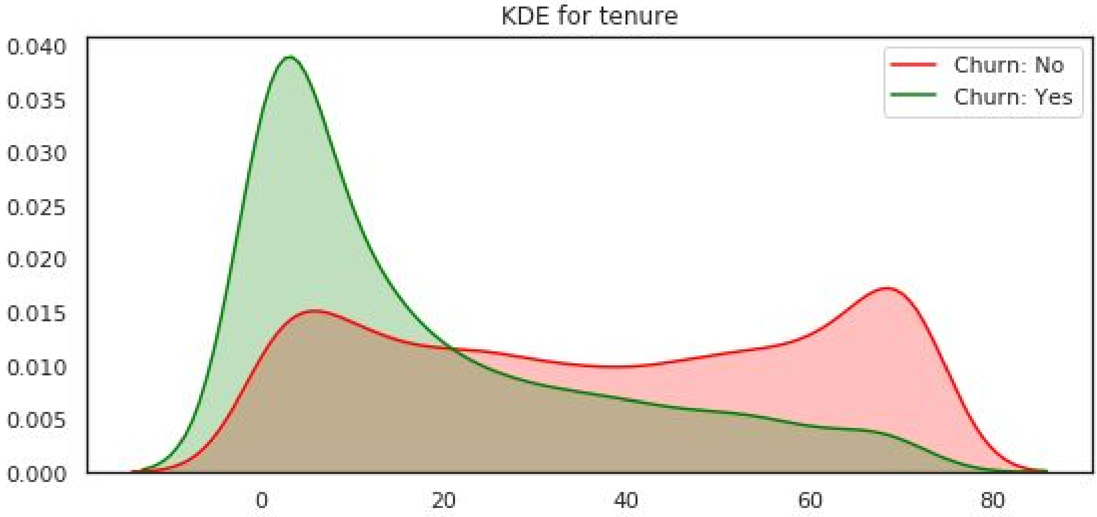
Let's look at customer attributes next.

<u>Tenure.</u>

First we'll plot tenure (membership length) vs number of customers.



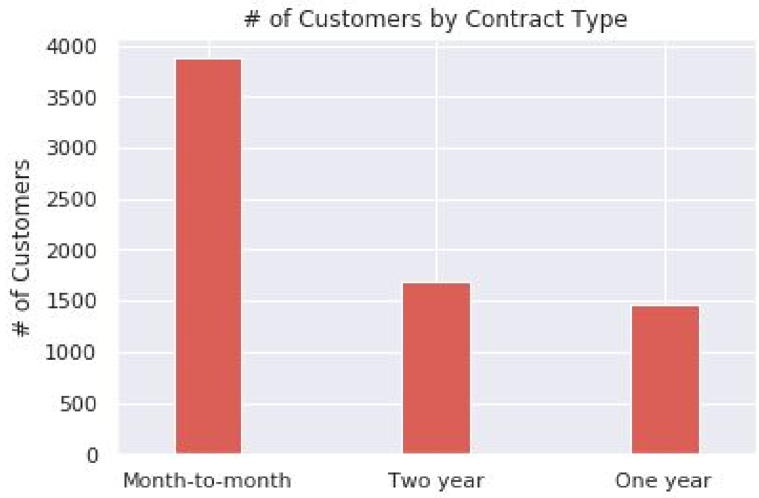
We can see most customers have either been with the business less than 10 month or greater than 70. Let's plot a probability density for churning customers vs their tenure.



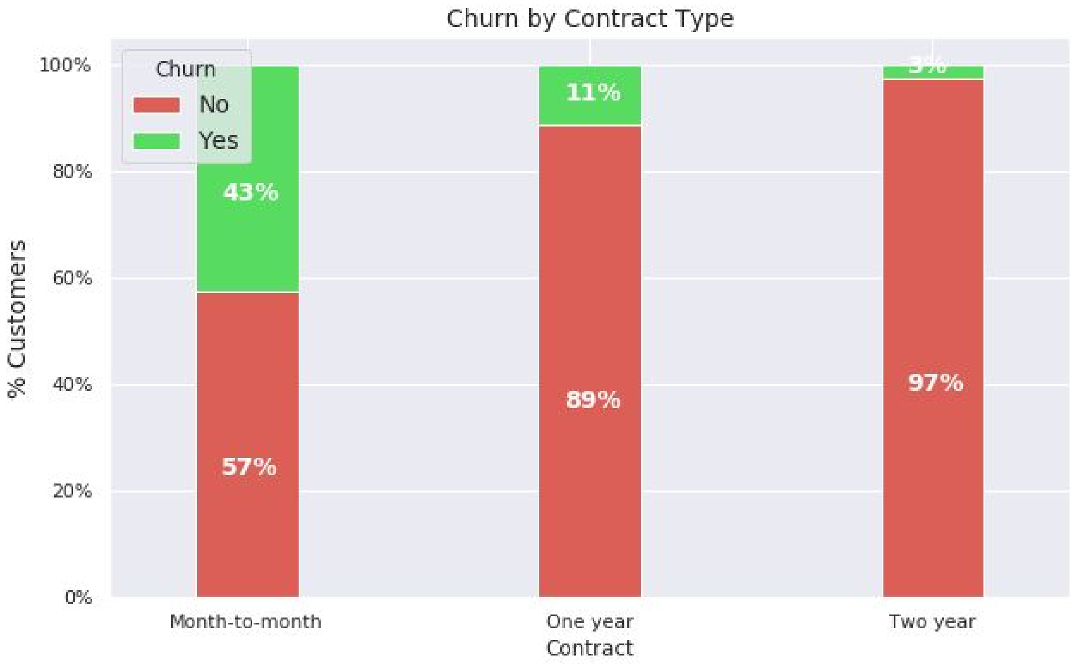
A large density of customers churn shortly after sign up and that tails of dramatically after 10 months of membership. The peak in non-churning customers is after 70 months but the distribution is relatively flat. These distributions may be down to contract length so let's examine that next.

Contract Length

First we'll plot the number of customers by contract type.



Most customers are on a month-month contract. Let's check churn per contract type.



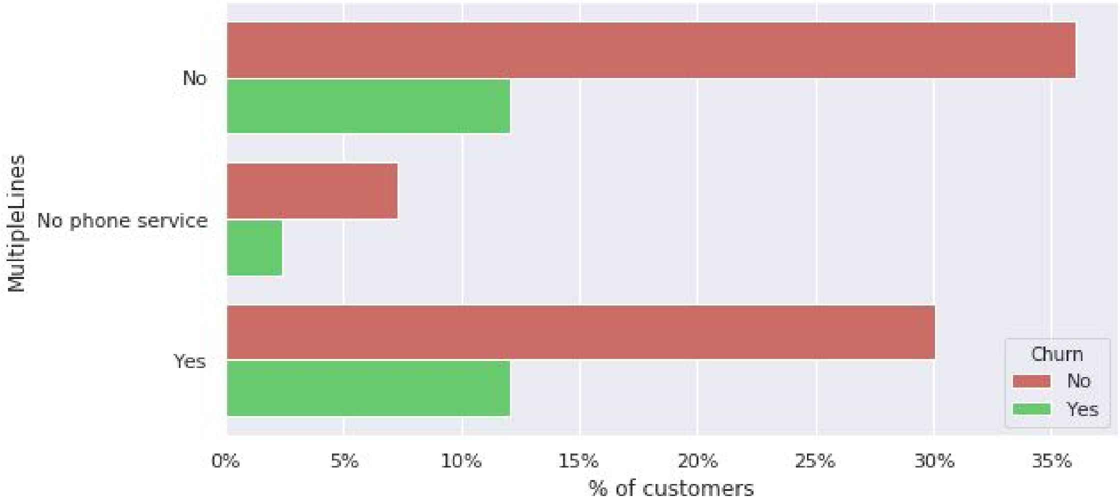
43% of customers on a month to month contract churn compared to only 11% and 3% respectively for one and two year contracts. This is a key insight to customer retention.

We can surmise that one and two year contracts probably have contractual fines and therefore customers wait until the end of their contract to churn. There is no data included with the dataset to validate this though.

Let's look at services next.

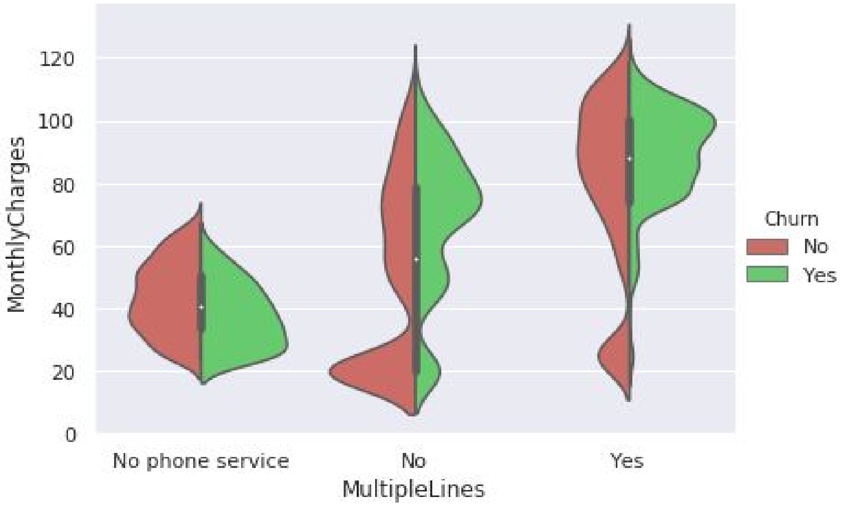
Phone Services

Here there are 3 attributes, customers either have no phone service, one line or multiple lines.



Customers with multiple lines have a slightly higher % churn rate than those with one line. Around 10% of customers have no phone service.

Plotting a violin graph of churn vs monthlycharges will reveal more info.



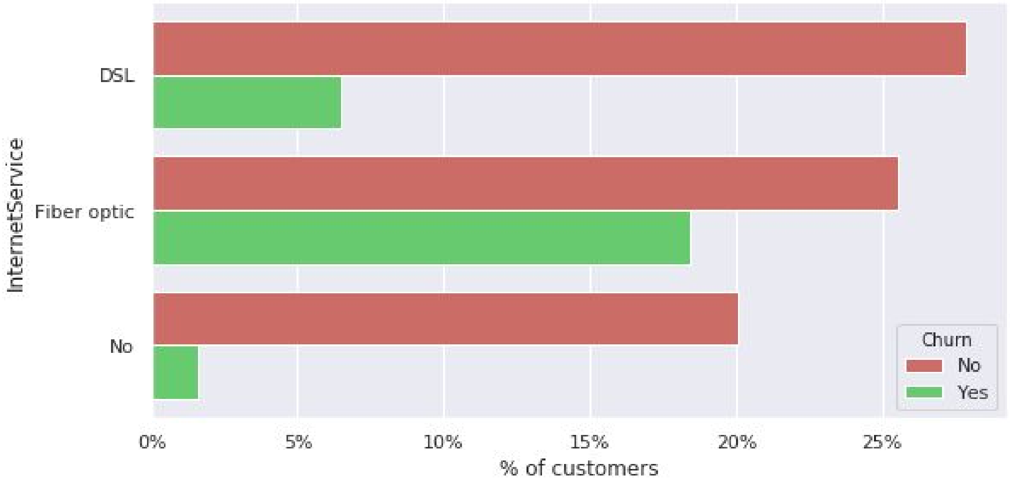
This is really interesting:

- For customers with no phone service the highest churning customers happen at a monthly charge (for broadband only) of around 30 dollars.
- For a single line customer 70 dollars is the point where most will churn
- For multi line customers 70 dollars is also a cut off point.

Let’s look at Internet Services next.

<u>Internet Services</u>

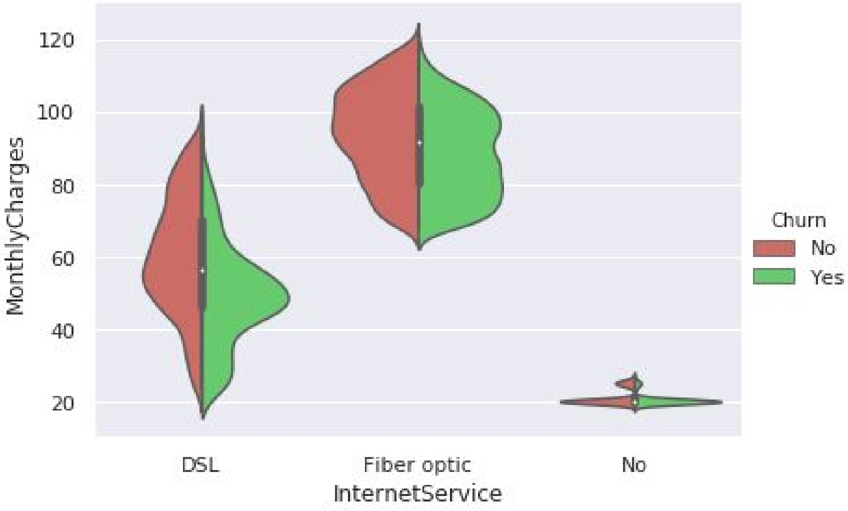
There are 3 attributes, customers either have no internet service, Fiber Optic or DSL.



We can see:

- Clients without internet (phone only service) have a very low churn rate
- Customers with fiber are more probable to churn than those with DSL connection

Plotting another violin graph of Internet service with monthly charges reveals:



For a DSL line, a price above 40 dollars causes high levels of churn.

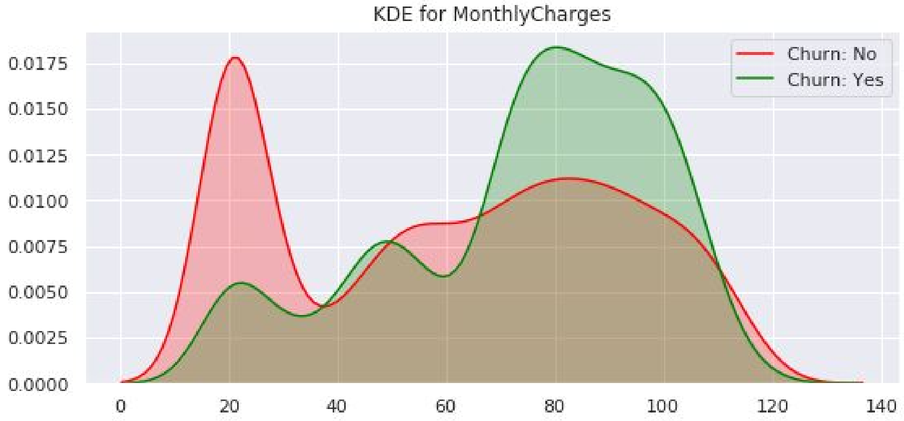
Customers with Fiber Optic services are prepared to pay a much higher service charge per month, over 60 dollars compared to DSL's 10 dollars. This would be good to look at to increase business revenues.

Customers with no internet services should not be on this chart, so some customers are paying for services they don't use.

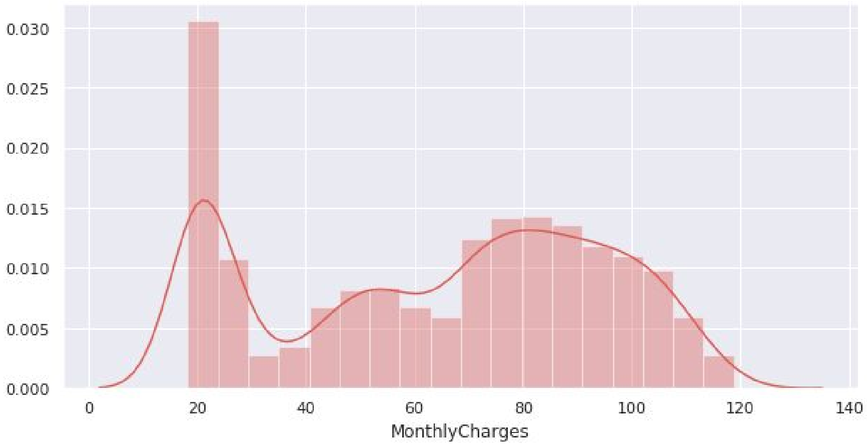
Let's look at Contracts and Payments next.

Contracts and Payments

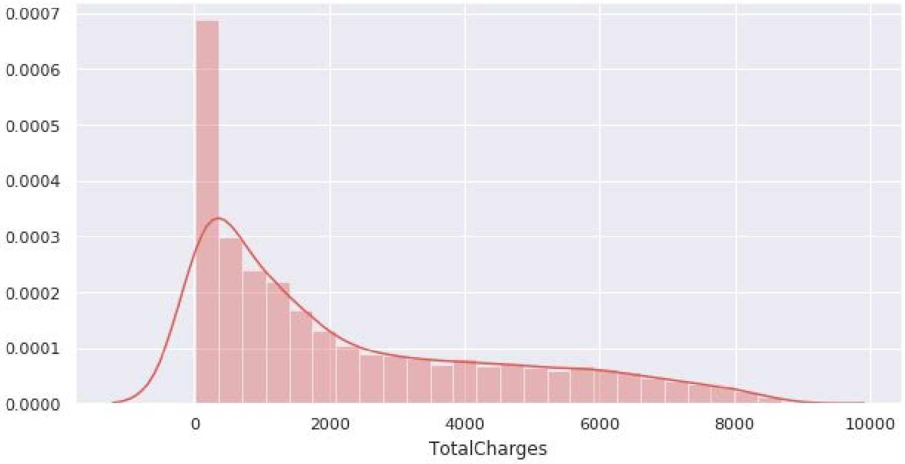
Plotting a probability density of monthly charges reveals:



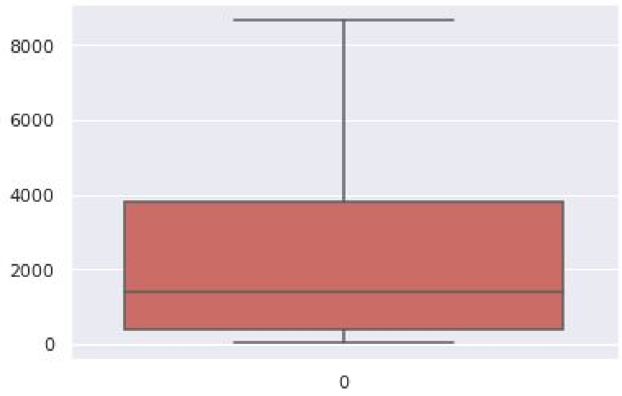
A large density of customers churn at the 75 dollar + per month price point. With customers paying around 20 dollars the most likely to stay. Let’s look at the distribution of charges.



30% of customers are charged around 20 dollars per month, these must be for no phone service (broadband only) customers. The next highest is 70-90 dollars per month. Let's look at the total charges per year.

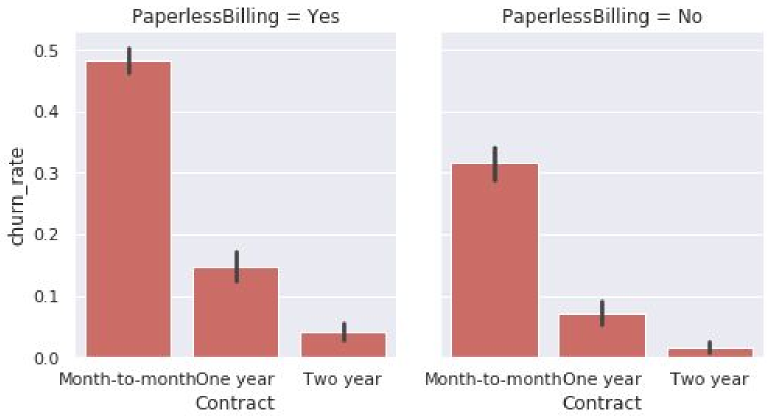


Over 65% of the customers pay less than 500 dollars per year. Plotting the probability density will reveal more about churning customers per yearly charge. Let's boxplot this to better visualise the attribute range.

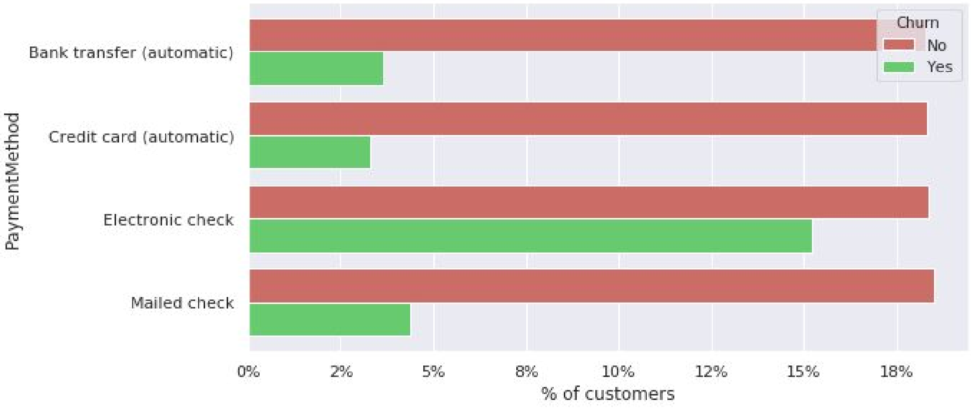


Most customers pay between 500 and 2000 dollars per year with the median being around 1500 dollars and the max being greater than 8000 dollars.

There are some further attributes we can plot: Paperless Billing and Payment vs Churn.



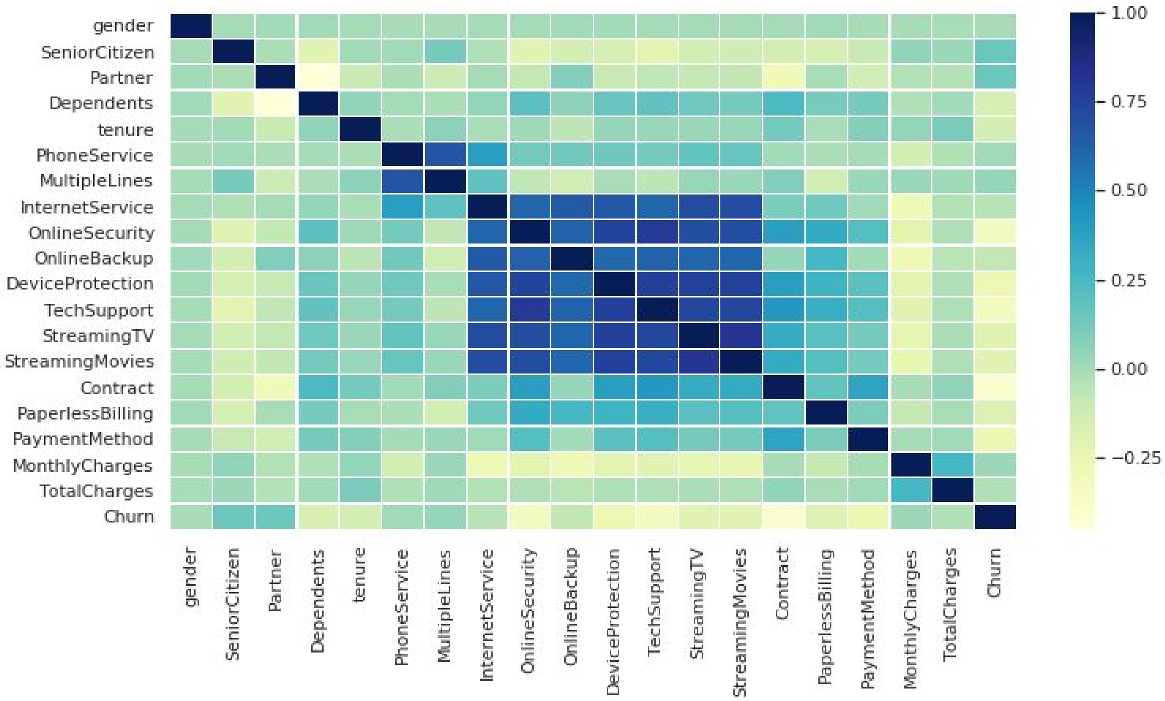
Regardless of whether the customer has paperless billing or not, customers on month to month contracts churn the most with the churn rate higher for paperless billing. Plot churn vs Payment.



From this graph we can see that the preferred payment method is Electronic check with around 35% of customers in total. This method also has a very high churn rate. The 3 other payment types have similar, low (<4%) churn rates.

Feature Importance

As a final visual let's plot a correlation heatmap to see if more data can be extracted.



In the center of the map are all the add on services, they are naturally correlated together. Looking out further the phone lines are also heavily correlated. Focusing on the churn row highlights that Senior Citizens, Partners, Phone Service, MultipleLines and MonthlyCharges are all correlated with churn, further reinforcing our visualisation insights.

Algorithms and Techniques

I intend on training the dataset on a variety of SciKit Learn classifiers and an AWS XGBoost classifier, well known for classification problems to see which give the best result. The following classifiers will be used with their default hyperparameters to start then tuned later depending on the result:

- Decision Tree
- Random Forest
- Gaussian Naive Bayes
- K Nearest Neighbors
- SVM
- Neural Network
- Logistic Regression
- Extra Trees
- AdaBoost
- Gradient Boosting
- XGBoost

The best 3 classifiers will be put into an ensemble classifier to improve the score.

The best classifier will then be tuned using GridSearchCV to improve the result further (if possible).

Because this is a Machine Learning course mainly focused on AWS development I will create a second independent notebook on the dataset on an S3 hosted bucket running a Linear Learner classifier. The hyperparameters will then be tuned to improve the score.

Benchmark

I could not find a similar reference project online but all companies strive to get customer churn to a minimum (as it affects their revenue). We will aim for 75% accuracy on our learning model, then tune the model to lower mis-classified customers.

III. Methodology

Data Preprocessing

For both training notebooks the dataset was preprocessed in a few ways.

First, as per the data exploration, the TotalCharges were converted to numerical data and the missing values imputed.

Second the CustomerID was dropped from the dataset as it has no effect on the classifier.

Third, the 'Churn' predictor variable was converted from yes/no to binary numeric for classification.

Lastly the non-numeric variable values were converted to numeric.

For the **SciKit learn** notebook, two datasets were created; one of the variable 'Churn' (y) and another with the rest of the data (X). The X dataset was then scaled using a MinMaxScalar and both datasets were split into test and train datasets for further processing.

For the **Linear Learner** notebook, 2 datasets were created; one called 'features' with the 'Churn' variable data and the other called 'labels with the rest of the data. Both datasets were converted to float32 types. The train_test_split import was called so we could split the 'features' 'labels' datasets into train and test sets.

The training data (features_train and labels_train) were converted from numpy to tensors using the write_numpt_to_dense_tensor so the Linear Learner can process it. The data was then uploaded to an S3 bucket using the AWS SDK boto3 implementation.

Implementation

For the **SciKit Learn notebook** the classifiers were input into an array ‘classifiers’.

A new dataframe (‘predictions_df’) was created and a new column of the Churn test data (y_test) was added to compare when scoring the results.

As each classifier is running through the same process and data, a for loop was created. Each classifier was then trained (fit) on the training data (X_train, y_train). After a new variable ‘predictions’ was created to use the trained data to predict on the test data (X_test). The predictions_df data frame was called to get the name of the classifier and set to the predicted data. The ‘predictions’ variable was then called to score it against the test data (y_test) to produce an accuracy score.

The top 3 classifiers were:

- GradientBoosting (0.809)
- Logistic Regression (0.807)
- XGBoost (0.779)

These were input into an ensemble classifier. Here I imported the VotingClassifier from sklearn.ensemble and set each classifier to a variable. The VotingClassifier was then called and each variable was fit using the estimator parameter. Soft voting was selected because we are looking for the best result (argmax) from the predicted probabilities. The result was then trained on the training data (X_train, y_train) using the VotingClassifier variable. A ‘predictions’ variable was created and used to create predictions based on the test data (X_test). The accuracy was then scored.

The ensemble accuracy was 0.805 which was lower than the GradientBoosting classifier (0.809). The GradientBoosting classifer will be the selected classifier to tune (see refinement below).

For the **Linear Learner** notebook a container containing the linear learner algorithm was created by calling the get_image_uri.

A new variable ‘linear; was then created containing the training parameters. The container was first called, followed by ‘role’ used to execute the current Sagemaker role. We used a ‘ml.c4.xlarge’ instance and trained it once by calling train_instance_type and train_instance_count respectively. An output location was specified and a sagemake_session called.

The model was trained on the uploaded training data by calling the ‘fit’ method and deployed to a ‘ml.m4.xlarge’ instance. The results varied depending on the hyperparameters chosen (see below).

Refinement

For the **SciKit learn** notebook, GridSearchCV was imported from sklearn so we can tune the GradientBoost classifier. Four parameters were chosen; max_depth_range, min_leaf_range, n_estimators and max_features and a range of starting, finishing and the steps to take were chosen. I varied the start and finish variables during training to find the best result. The variables were then put into a dictionary called 'param_grid'.

A new variable 'grid' was called which contained the GridSearchCV algorithm and it's parameters. 'GBClass' is the GradientBoosting classifier, 'param_grid' is the above variables, 'cv' (cross validation variable) was chosen as it's default of 5, 'scoring' of accuracy was chosen as we want to know how the result scored, 'verbose=10' for 10 messages and 'return_train_score=True' for viewing the under/over fitting parameters.

After training a few times, optimal parameters were produced. I retrained the GradientBoost with these parameters and fit them on the training data (X_train, y_train). An accuracy of 0.814 was obtained.

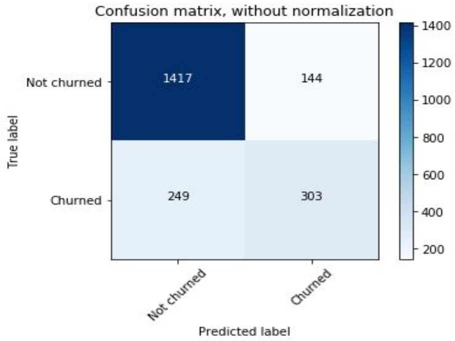
For the **Linear Learner** notebook, four parameters were chosen and varied. 'Feature_dim' = 1 as we only have one feature (churn). 'Predictor_type' = binary_classfication; either the customer churns or doesn't. 'mini_batch_size' = 200; number of observations per batch and a 'learning_rate' which varied from 0.1 to 0.0001.

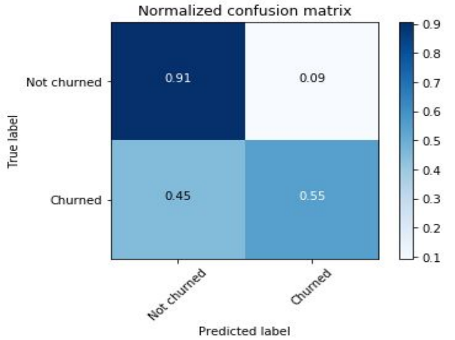
After tuning and deployment the model was scored against the test data and an accuracy of 0.810 was achieved.

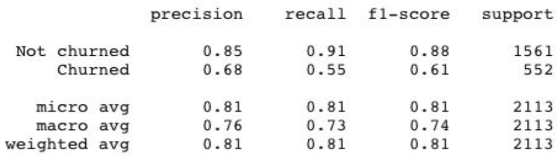
IV. Results

Model Evaluation and Validation

To evaluate the models further I plotted both into a confusion matrix to check for precision and recall. The GradientBoosting model scored:

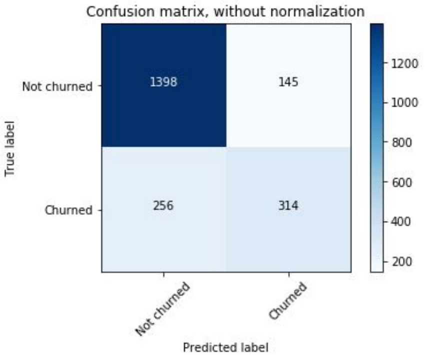


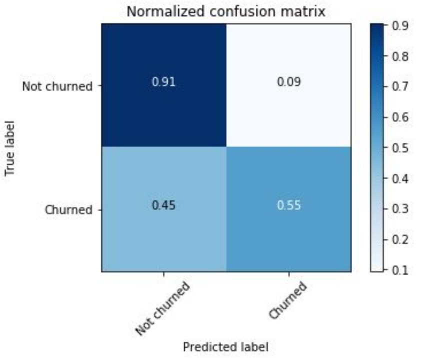


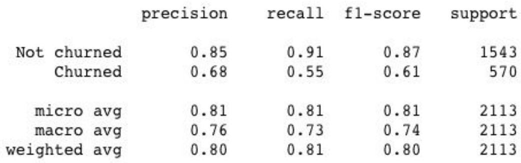


```
             precision    recall  f1-score   support

 Not churned       0.85      0.91      0.88      1561
     Churned       0.68      0.55      0.61       552

   micro avg       0.81      0.81      0.81      2113
   macro avg       0.76      0.73      0.74      2113
weighted avg       0.81      0.81      0.81      2113
```

Whilst the Linear Learner scored:







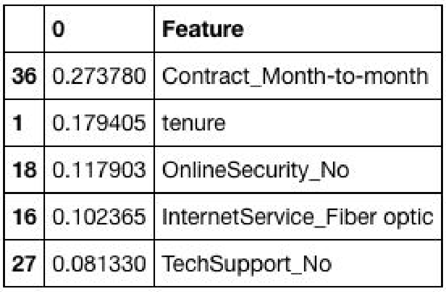
```
             precision    recall  f1-score   support

 Not churned       0.85      0.91      0.87      1543
     Churned       0.68      0.55      0.61       570

   micro avg       0.81      0.81      0.81      2113
   macro avg       0.76      0.73      0.74      2113
weighted avg       0.80      0.81      0.80      2113
```

Justification

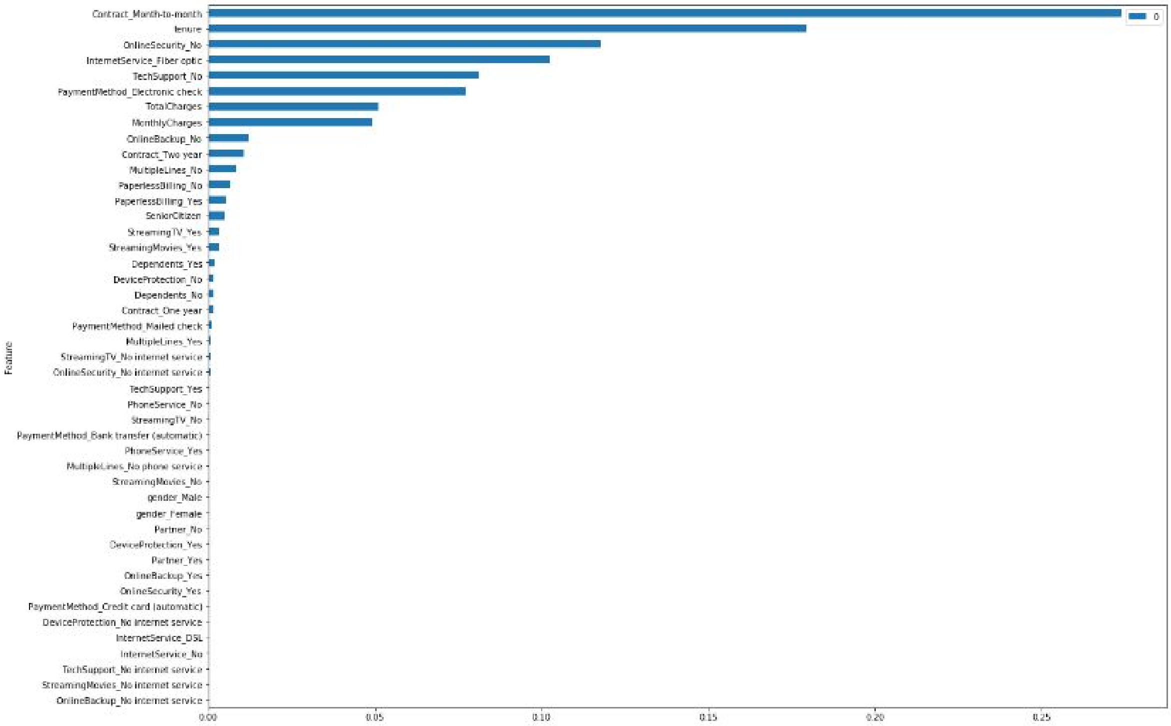
Both models scored over 80% which is higher than our 70% initial target.

Both models had the same precision and recall but the GradientBoosting classifier, classified more true positives and true negatives and misclassified less false positives and negatives. It also scored marginally higher too. We will use the GradientBoosting classifier for our final results.

Predicting the most important features that affect churn involves calling the ‘feature_importances_’ method on our classifier. We can then visualise the results in tabular and graph form:

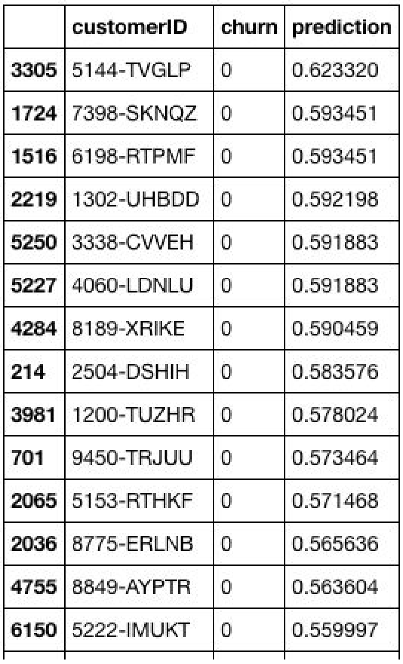


	0	Feature
36	0.273780	Contract_Month-to-month
1	0.179405	tenure
18	0.117903	OnlineSecurity_No
16	0.102365	InternetService_Fiber optic
27	0.081330	TechSupport_No



From this we can see that top 5 features affecting churn were similar to ones highlighted in the data analysis phase. The feature that most affects customer churn is being on a month to month contract.

We can take our churn predictions one stage further and actively predict who will be the next customer to churn. By creating 3 datasets, one for churn, one for customerID and one for the rest of the features we can predict the probabilities of customer churn using the ‘predict_proba’ method. We set churn=0, to select customers who have not yet churned and sort based on prediction, with the highest first.



	customerID	churn	prediction
3305	5144-TVGLP	0	0.623320
1724	7398-SKNQZ	0	0.593451
1516	6198-RTPMF	0	0.593451
2219	1302-UHBDD	0	0.592198
5250	3338-CVVEH	0	0.591883
5227	4060-LDNLU	0	0.591883
4284	8189-XRIKE	0	0.590459
214	2504-DSHIH	0	0.583576
3981	1200-TUZHR	0	0.578024
701	9450-TRJUU	0	0.573464
2065	5153-RTHKF	0	0.571468
2036	8775-ERLNB	0	0.565636
4755	8849-AYPTR	0	0.563604
6150	5222-IMUKT	0	0.559997

These are the customers we should be targeting, along with the top5 features to reduce customer churn rate.

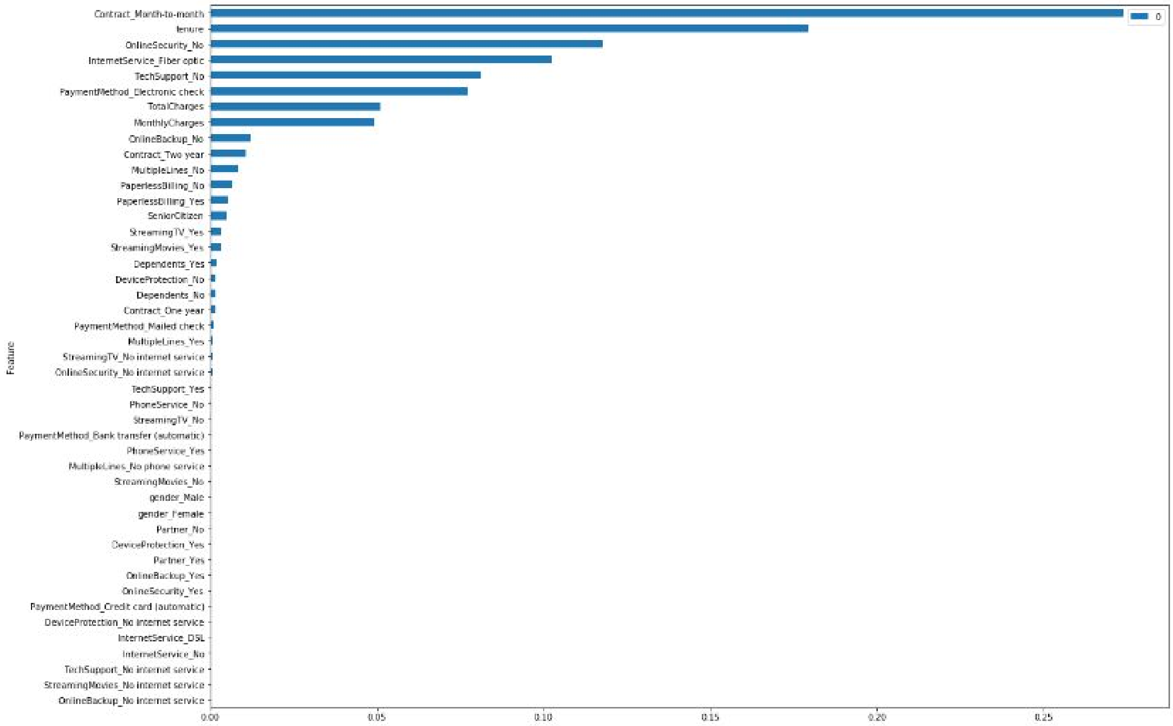
V. Conclusion

Free-Form Visualization

From the 19 features of data (excluding churn - our target variable and customerID) we have been able to provide management with a strategy to reduce churning customers in the business.

The graph below shows which feature we should target first; reducing the number of month-month contracts offered. It also shows there are 7 top features that reduce churn (tenure is listed as number two but is ambiguous as we want to increase the number of months a customer has been with us - see improvement below) that the Telco management should focus on in the future.

The graph below also lists ALL features (as the dataset was turned into a ‘dummies’ dataset) which provides further detail for customer retention as the features that don’t impact churn - impact retention, providing management with further tactics. Management could even prioritise to only signup new customers with these ‘retention’ features as they know they already offer a service that matches these customers requirements.



Reflection

The project involved analysing a dataset of 7043 Telecommunication customers to find which customers would churn and what are the features that cause them to churn. We were provided with specific features about the customers from gender to dependents, to the services they purchase and the monthly and year prices they pay.

The first step was to analyze the data. To look for any missing values, the type of data provided and whether it was correctly categorised. After we charted specific, individual attributes of the customers to see how churning and non churning customers were affected by the features provided.

We then trained various classifiers from SciKit Learn on the data. The best classifiers were input into an ensemble classifier to improve the results and it was found the GradientBoost classifier scored best for accuracy. We tuned the model with GridSearchCV to get the best accuracy we could.

We created another notebook to train a Linear Learner classifier to see if we could improve on the GradientBoosting classifier. The accuracy of the 23 classifiers were very similar in accuracy, precision and recall but the GradientBoosting classifier was just slightly better.

We used this classifier to predict the features affecting churn and customers who were predicted to churn next.

I found it difficult to get the overall accuracy of the model over 80.1% even after tuning. I also found it difficult to get the customer prediction data over 55%.

The final model fit my expectations for features that affect customer churn. Many of those features were identified during data analysis. My expectations for customer predictions proved inconclusive. Although this model gives the business a good foundation of customers to target we really need to get the score over 65% to be more certain they will actually churn.

Improvement

The accuracy of the model may be a limitation of the dataset. The best way to test this would be to increase the number of customers from the Telco business and rerun the results. This

We could also eliminate some features that we know does not affect churn to see if it improves the result further.

We trained on a variety of algorithms, so choosing different ones may provide us with a slightly improved result but it would only be marginal. The best algorithms were also tuned to the best parameters available.

We could use the new model as the new benchmark for choosing features to eliminate churn but I would not use it as the benchmark for prediction exactly which customers will churn until we have gathered more data and rerun the results.