Final Report

January 29th, 2019

Regarding:

Predicting Customer Churn

Cliff Robbins
cliff@gearforgesoftware.com
M 612.701.2998

Introduction

The project will focus on a problem that 28 million business face each day of operation, customer churn.

Definition

Customer churn, also known as customer attrition, customer turnover or customer defection is the loss of clients or customers. Many companies include customer churn rate as part of their monitoring metrics because the cost of retaining current customers compared to acquiring new customers is much less.
Within customer churn there is the concept of voluntary and involuntary churn with voluntary being a customer leaves on their own choice while involuntary could be attributed to customer relocation to a long term care facility, death or customer relocation in a different state/geography. In most analytical models, involuntary churn is excluded from the metric.

Formulation of a Question

When a company first starts up, the founding members can typically handle all of the various customer concerns. As the company continues to grow, the founders can no longer service all of the various clients with support handled by a customer service team. The customer service team focuses on current issues and a proactive approach is lost.
As the company grows, the company still cares about its clients; however, due to the large customer base they can no longer address each and every customer. This is a real problem for companies. How does a company proactively predict if a customer is happy or unhappy? How does a company know if a customer is so unhappy that they are willing to leave? If a company knew if a customer was getting ready to leave, could they reach out to the customer and mend the relationship?

Hypothesis

I believe past customer data can predict future customer churn.

Prediction

If I had past customer data that showed various features and whether they stayed or churned we could use that data to predict future outcomes of current customers.

Testing

To test my hypothesis, I will use a set of customer data with various features along with whether they churned or not.

The data has 7043 rows and can be found at:

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

The dataset has the following features:

- 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

The following target will be used to understand if the customer churned or not.

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

Analysis

To determine if we can predict the churn rate, I will use various classification algorithms, and will compare them according to the appropriate performance metrics.

Approach

Data Acquisition and Wrangling

Data Investigation

The first step was to import the data and the investigate the data.

The data is located in a csv file which I imported into a Panda's DataFrame using the read_csv function. I have the data stored in a subfolder under the Jupyter notebook so others can leverage the same data set.

After importing the data, I ran a head function to show the first 5 rows to understand what the data looked like.

I then started looking for missing values.

1. I started initially looking for any null values by column. My dataframe came back with zero null values.
2. I then looked for any empty strings by row. My results returned 11 rows that had empty strings.

Data Cleaning

Once I understand what columns had issues, I also wanted to understand if Pandas had assigned the correct types to each column. I ran a .info method and it showed almost all columns were set to object. This meant I needed to get a better understanding of each column data type.

Based on the head method, I then listed out each column that I felt was categorical using the unique method and converting them to a list to see the unique values. Here is the printout:

```
gender:  ['Female', 'Male']
SeniorCitizen:  [0, 1]
Partner:  ['Yes', 'No']
Dependents:  ['No', 'Yes']
tenure:  [1, 34, 2, 45, 8, 22, 10, 28, 62, 13, 16, 58, 49, 25, 69, 52, 71,
21, 12, 30, 47, 72, 17, 27, 5, 46, 11, 70, 63, 43, 15, 60, 18, 66, 9, 3,
31, 50, 64, 56, 7, 42, 35, 48, 29, 65, 38, 68, 32, 55, 37, 36, 41, 6, 4,
33, 67, 23, 57, 61, 14, 20, 53, 40, 59, 24, 44, 19, 54, 51, 26, 0, 39]
PhoneService:  ['No', 'Yes']
MultipleLines:  ['No phone service', 'No', 'Yes']
InternetService:  ['DSL', 'Fiber optic', 'No']
OnlineSecurity:  ['No', 'Yes', 'No internet service']
OnlineBackup:  ['Yes', 'No', 'No internet service']
DeviceProtection:  ['No', 'Yes', 'No internet service']
TechSupport:  ['No', 'Yes', 'No internet service']
StreamingTV:  ['No', 'Yes', 'No internet service']
StreamingMovies:  ['No', 'Yes', 'No internet service']
Contract:  ['Month-to-month', 'One year', 'Two year']
PaperlessBilling:  ['Yes', 'No']
PaymentMethod:  ['Electronic check', 'Mailed check', 'Bank transfer
(automatic)', 'Credit card (automatic)']
```

Based on this, I decided that all of them except tenure would be set to a type of category.

I had also noticed that TotalCharges was an object and not a float64, which made me suspicious that something wasn't right. When I investigated, it had 11 rows with empty strings. I looked at the 11 rows and could see that they data was 'off'.

Dealing with Missing Data Values

The only column that has missing values was the TotalCharges column. After looking at the 11 rows, the data looked invalid so I decided to fill in the 11 rows. I filled in the 11 rows with zero's and then assigned the column as type float64.

Data Outliers

After dealing with missing values and assigning the proper types, I then used the describe method so I could take a look at the numerical types and understand if I had any values that looked odd. Based on that readout, the values appear to me normal of what I would expect for monthly and total charges.

```
In [48]: #now lets see if we have any outliers
         assigned_customer_churn_df.describe()
```

Out[48]:

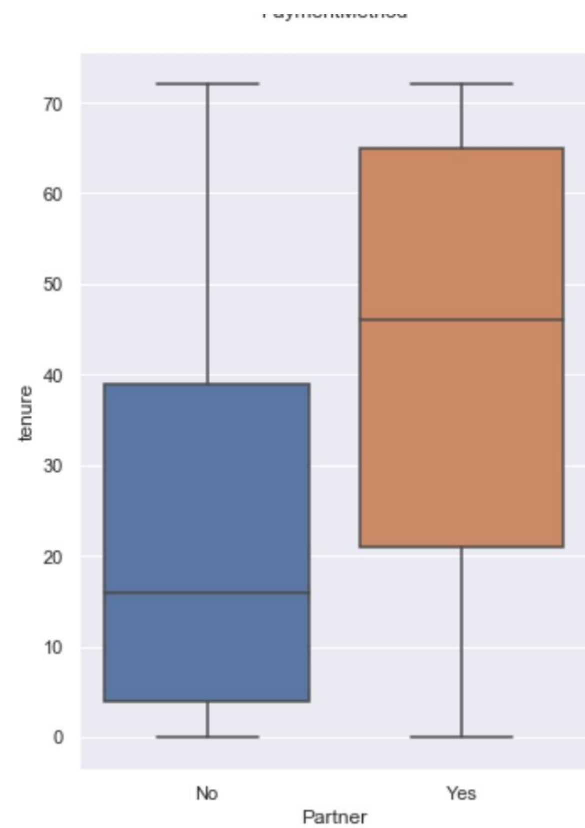
	tenure	MonthlyCharges	TotalCharges
count	7032.000000	7032.000000	7032.000000
mean	32.421786	64.798208	2283.300441
std	24.545260	30.085974	2266.771362
min	1.000000	18.250000	18.800000
25%	9.000000	35.587500	401.450000
50%	29.000000	70.350000	1397.475000
75%	55.000000	89.862500	3794.737500
max	72.000000	118.750000	8684.800000

Storytelling and Inferential Statistics

EDA Inferential Statistics Investigation

The first step of EDA was to visualize the data to understand the relationship between the various features and the predictor. I initially used tenure to understand how long a customer stayed hypothesizing that the longer a customer stayed the less likely they are to churn.

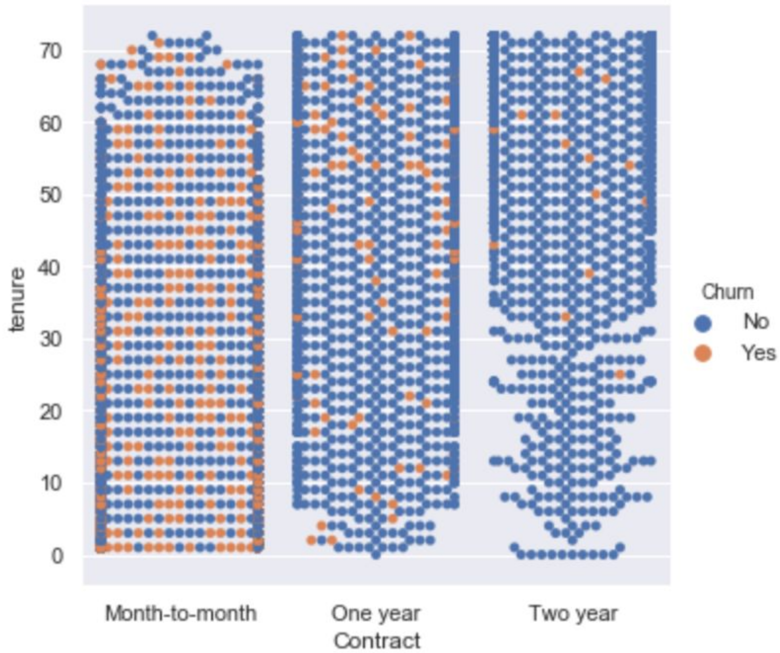
Because most of my data is categorical, I started with boxplots to understand the categories within each feature against tenure. What I noticed was that some of the categories within each feature had higher levels of tenure than their counterparts. Here is an example of Partner vs Tenure that shows those with a partner have a higher level of tenure.



The next step was to take the following features and do a catplot of the categories against tenure along with churn.

1. Phone Service
2. Multiple Lines
3. Internet Service
4. Contract Length
5. Paperless Billing
6. Payment Method
7. Dependents
8. Senior Citizen
9. Partner

Amongst those features, I could see distinct patterns that indicate some categories are prone to churn more than others. In the below graphic, you can see that month-to-month billing has more frequencies of churn than the other 2 contract types.



Leveraging Inferential Statistics

The features that provide visual correlations between the categories and churn next need to be checked for correlation strength.

I set Alpha equal to 0.05 or 5%.
My hypothesis is no relationship between categories and churn.

We will leverage p-value, Pearson Chi-Square and Cramer's phi.

Note: Cramer's phi will measure how strong the relationship between the 2 variables, the closer to 1 the strong the relationship.

I tested the following categorical features against churn:

- gender
- SeniorCitizen
- Partner
- Dependents
- PhoneService
- MultipleLines
- InternetService
- Contract

- PaperlessBilling
- PaymentMethod

Here are the results for all of the categorical features:

gender

<u>No Relationship (fail to reject H0)</u>

Comparison of: gender to Churn.

	Churn		
	No	**Yes**	**All**
gender			
Female	49.27	50.24	49.52
Male	50.73	49.76	50.48
All	100.00	100.00	100.00

	Chi-square test	**results**
0	Pearson Chi-square (1.0) =	0.5224
1	p-value =	0.4698
2	Cramer's phi =	0.0086

SeniorCitizen

<u>Relationship (reject H0)</u>

Comparison of: SeniorCitizen to Churn.

	Churn		
	No	**Yes**	**All**
SeniorCitizen			
0	87.13	74.53	83.79
1	12.87	25.47	16.21
All	100.00	100.00	100.00

	Chi-square test	**results**
0	Pearson Chi-square (1.0) =	160.3521
1	p-value =	0.0000
2	Cramer's phi =	0.1509

Partner

<u>Relationship (reject H0)</u>

Comparison of: Partner to Churn.

	Churn		
	No	**Yes**	**All**
Partner			
Yes	52.82	35.79	48.3
No	47.18	64.21	51.7
All	100.00	100.00	100.0

	Chi-square test	**results**
0	Pearson Chi-square (1.0) =	159.4145
1	p-value =	0.0000
2	Cramer's phi =	0.1504

Dependents

<u>Relationship (reject H0)</u>

Comparison of: Dependents to Churn.

	Churn		
	No	**Yes**	**All**
Dependents			
No	65.52	82.56	70.04
Yes	34.48	17.44	29.96
All	100.00	100.00	100.00

	Chi-square test	**results**
0	Pearson Chi-square (1.0) =	189.9403
1	p-value =	0.0000
2	Cramer's phi =	0.1642

PhoneService

<u>No Relationship (fail to reject H0)</u>

Comparison of: PhoneService to Churn.

	Churn		
	No	**Yes**	**All**
PhoneService			
No	9.9	9.1	9.68
Yes	90.1	90.9	90.32
All	100.0	100.0	100.00

	Chi-square test	**results**
0	Pearson Chi-square (1.0) =	1.0044
1	p-value =	0.3162
2	Cramer's phi =	0.0119

MultipleLines

Relationship (reject H0)

Comparison of: MultipleLines to Churn.

	Churn		
	No	**Yes**	**All**
MultipleLines			
No phone service	9.90	9.10	9.68
No	49.11	45.43	48.13
Yes	40.99	45.48	42.18
All	100.00	100.00	100.00

	Chi-square test	**results**
0	Pearson Chi-square (2.0) =	11.3304
1	p-value =	0.0035
2	Cramer's V =	0.0401

InternetService

Relationship (reject H0)

Comparison of: InternetService to Churn.

	Churn		
	No	**Yes**	**All**

InternetService			
DSL	37.92	24.56	34.37
Fiber optic	34.77	69.40	43.96
No	27.31	6.05	21.67
All	100.00	100.00	100.00

	Chi-square test	results
0	Pearson Chi-square (2.0) =	732.3096
1	p-value =	0.0000
2	Cramer's V =	0.3225

Contract

Relationship (reject H0)

Comparison of: Contract to Churn.

	Churn		
	No	**Yes**	**All**
Contract			
Month-to-month	42.91	88.55	55.02
One year	25.26	8.88	20.91
Two year	31.83	2.57	24.07
All	100.00	100.00	100.00

	Chi-square test	results
0	Pearson Chi-square (2.0) =	1184.5966
1	p-value =	0.0000
2	Cramer's V =	0.4101

PaperlessBilling

Relationship (reject H0)

Comparison of: PaperlessBilling to Churn.

	Churn		
	No	**Yes**	**All**
PaperlessBilling			
Yes	53.56	74.91	59.22

No	46.44	25.09	40.78
All	100.00	100.00	100.00

	Chi-square test	**results**
0	Pearson Chi-square (1.0) =	259.1610
1	p-value =	0.0000
2	Cramer's phi =	0.1918

PaymentMethod

<u>Relationship (reject H0)</u>

Comparison of: PaymentMethod to Churn.

	Churn		
	No	**Yes**	**All**
PaymentMethod			
Electronic check	25.01	57.30	33.58
Mailed check	25.20	16.48	22.89
Bank transfer (automatic)	24.86	13.80	21.92
Credit card (automatic)	24.93	12.41	21.61
All	100.00	100.00	100.00

	Chi-square test	**results**
0	Pearson Chi-square (3.0) =	648.1423
1	p-value =	0.0000
2	Cramer's V =	0.3034

Initial EDA and Inferential Statistics Recap

Based on my visual EDA I had anticipated that all of the following features had correlation or relationships between the categorical feature and churn; however, I was wrong.

- gender
- SeniorCitizen
- Partner
- Dependents
- PhoneService
- MultipleLines

- InternetService
- Contract
- PaperlessBilling
- PaymentMethod

What I found was that the gender and phone service did not have a relationship with churn which is contrary to what I expected.

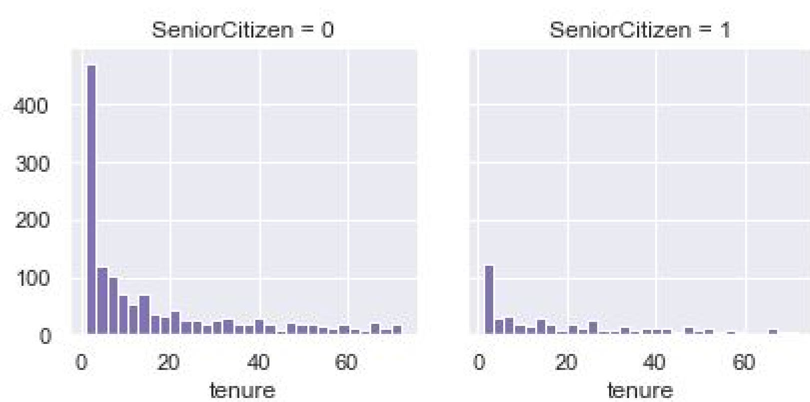
The dataset has 18 features and 1 target used to determine if the customer churned or not. Of the 18 features, 7 features have a relationship with churn.

- SeniorCitizen
- Partner
- Dependents
- InternetService
- Contract
- PaperlessBilling
- PaymentMethod

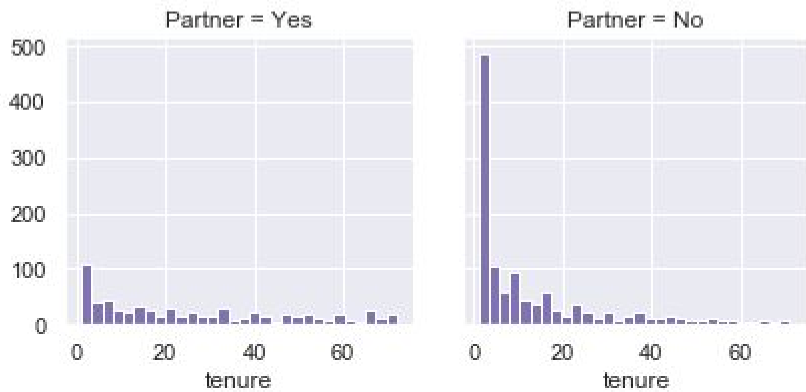
Visual EDA of 7 Features

Based on the 7 features identified above, we can now do a visual EDA to understand the frequency of the churned customers. The below visuals are taken from a subset of the data of only churned customers compared to tenure.

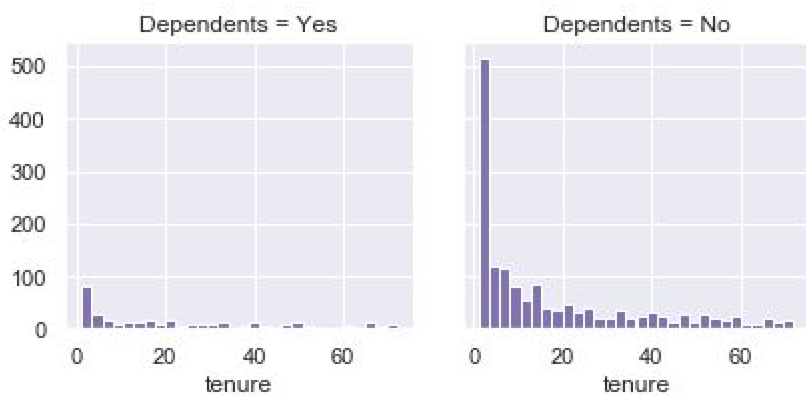
Senior Citizen



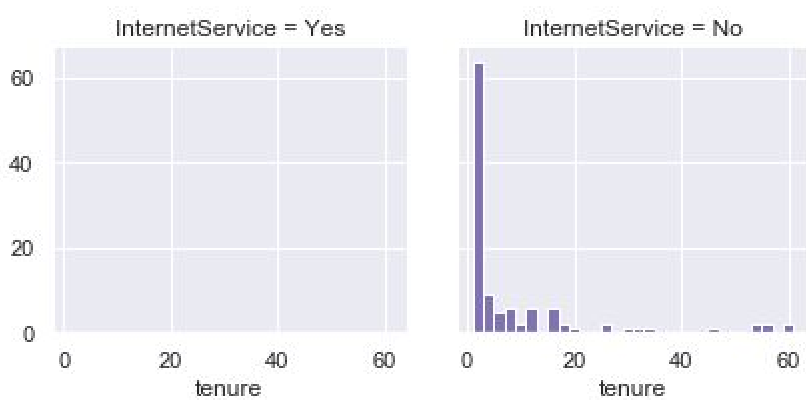
Partner



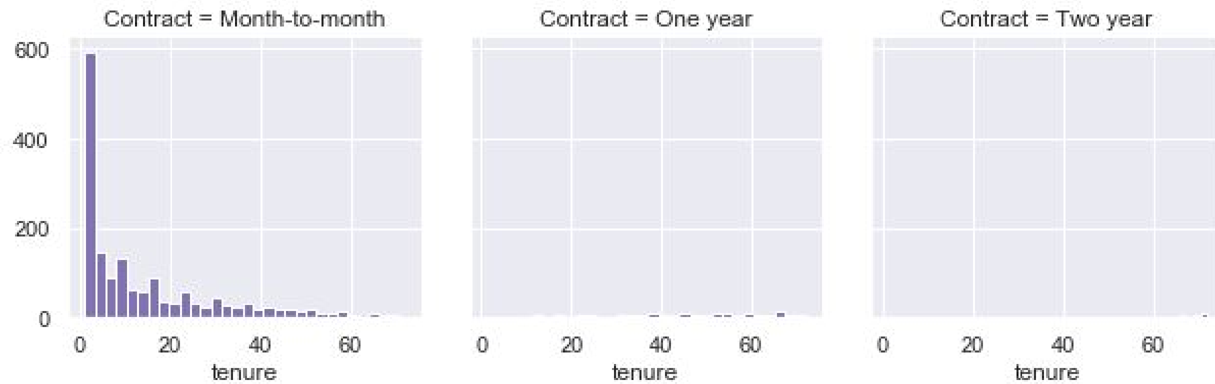
Dependents



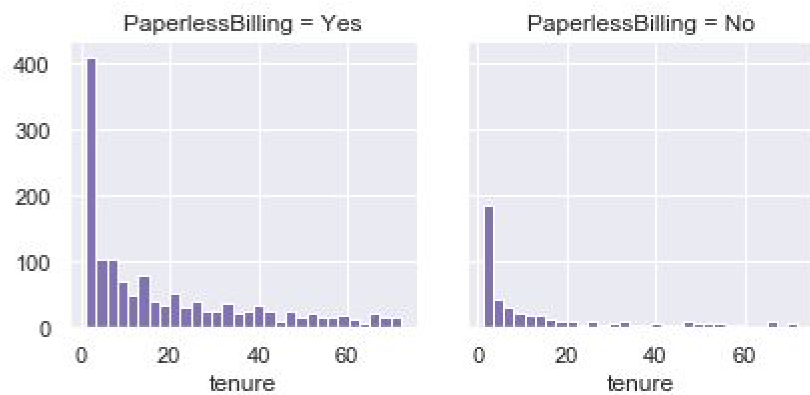
Internet Service



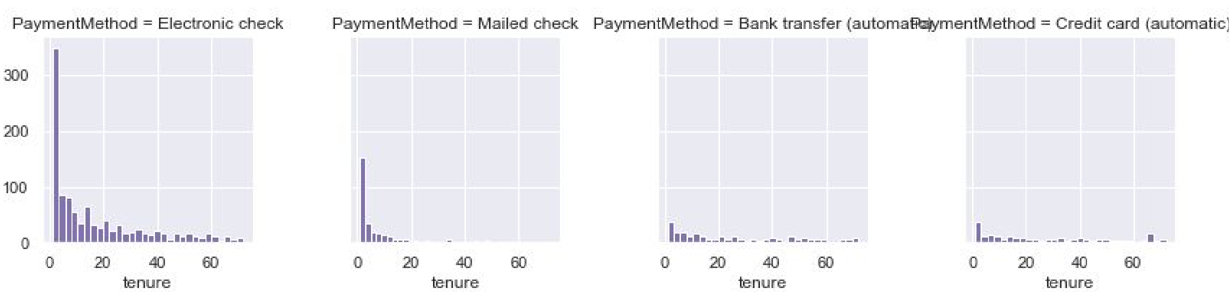
Contract



Paperless Billing



Payment Method



Visual EDA Recap of 7 Features

Based on the 7 feature comparisons, we can see some very distinct patterns emerge. Each of the services when broken down by their distinctive parts now show customers that have a higher rate of churn.

The following scenarios have a much higher rate of churn:

- Non-Senior Customers are more likely to churn in the first few years.
- Customers without partners are more likely to churn in the first few years.
- Customers without dependents are more likely to churn in the first few years.
- Customers without internet service are more likely to churn in the first few years.
- Customers that are on month-to-month billing are more likely to churn in the first few years.
- Customers that receive paperless billing are more likely to churn in the first few years.
- Customers that pay via Electronic Check are more likely to churn in the first few years.

Baseline Analysis and Preliminary Results

The goal of the baseline analysis is to establish a starting point and then improve from there. The evaluation of the model will use accuracy along with performance metrics.

The overall goal is to predict customer churn which means we are more interested in predicting if a customer might churn. That means we are okay if our false positive rate is high because we are more interested in not missing a customer that might churn.

The metric I will focus on is the Recall rate for the True classification. Recall rate is reported in the classification report that uses the confusion matrix. A higher recall rate for True means the model is predicting the correct customers that might churn (remember, we are okay if we have false positives).

Logistic Regression

The baseline model used logistic regression with L1 and L2 optimization using the following features:

- SeniorCitizen
- Partner
- Dependents
- InternetService
- Contract
- PaperlessBilling
- PaymentMethod

The logistic regression model also has various hyperparameters and I will focus on getting the optimal C value for L1 and L2. After testing various values, the optimum C values are:

L1	0.1
L2	0.01

Running the logistic regression model with those C values produced the following accuracy scores.

Logistic Model	**Training Data**	**Test Data**
L1	0.751419916698	0.743327654742
L2	0.751041272245	0.733674048836

The accuracy rate for the baseline after only optimizing the C value is roughly 75%. That is not bad for accuracy without much optimization. The next step is to run the classification report, here are the results.

L1 Training	**Precision**	**Recall**	**F1-Score**
False	0.74	0.74	0.74
True	0.27	0.27	0.27
Avg/Total	0.61	0.61	0.61

L1 Test	**Precision**	**Recall**	**F1-Score**
False	0.73	0.73	0.73
True	0.24	0.24	0.24
Avg/Total	0.60	0.60	0.60

L2 Training	**Precision**	**Recall**	**F1-Score**

False	0.74	0.74	0.74
True	0.27	0.27	0.27
Avg/Total	0.61	0.61	0.61

L2 Test	**Precision**	**Recall**	**F1-Score**
False	0.73	0.72	0.73
True	0.26	0.27	0.27
Avg/Total	0.61	0.60	0.61

The recall rate for L1 and L2 for the True classification is only ~0.27 which is low compared to the recall rate for False classification. That means customer churn will get predicted correctly 27% of the time which is not acceptable.

What the classification report is demonstrating is a natural imbalance in the data. The data has more False customer churns than True which is reflected in the model. The next step is to try resampling techniques along with different models.

Extended Analysis and Final Results

In the baseline analysis, I fit the data to a Logistic Regression model using L1 and L2 regularization. The accuracy was 75% for L1 and 74% for L2. The performance of the model showed an imbalance regarding customers that churned. The F1 score for customers that did not churn was 74% and the F1 score for customers that did churn was 27% (for L1 regularization).

The main score I am optimizing is the performance recall for the 'True' class. Because the client is more concerned about catching customers before they churn it is okay to have false positives. When using the base Logistic Regression recall is 93% for 'False' and 23% for 'True'.

This is indicative of the data set where the customers that churned have a much lower percentage compared to those that did not churn. This is a balance classification issue which cannot be fixed with throwing more data at it because there is a natural imbalance between the classes.

In this section I will use different models and data sampling techniques to test if the accuracy and/or performance improves.

SMOTE

The first sampling technique I will use is SMOTE. SMOTE stands for Synthetic Minority Over-sampling Technique. This means it will use a synthetic technique to add samples to the minority class. After applying SMOTE, the shape of the data is changed.

	False	**True**
Original Data Set Shape	3880	1402
SMOTE Data Set Shape	3880	3880

RUS

The second sampling technique I use is RUS. RUS stands for Random Under-sampling. This means it will randomly undersample the dataset.

	False	**True**
Original Data Set Shape	3880	1402
RUS Data Set Shape	1402	1402

Hyperparameter Tuning

The various models used all have parameters that can be changed to improve the model. For each model, I adjusted 1 or 2 parameters for an initial model comparison.

Model Comparison

Original Dataset

Model	**Precision (True)**	**Recall (True)**	**Accuracy**
Logistic Regression	0.54	0.227	0.744
Naïve Bayes	0.51	0.642	0.741
Decision Tree	0.55	0.435	0.756

kNN	0.52	0.435	0.744
SVM	0.55	0.456	0.756
Random Forest	0.56	0.471	0.762
AdaBoost	0.57	0.452	0.764

ReSampled Dataset

Model	**Precision (True)**	**Recall (True)**	**Accuracy**
SMOTE - Logistic Regression	0.44	0.805	0.676
SMOTE - Naïve Bayes	0.46	0.773	0.699
SMOTE - Decision Tree	0.47	0.756	0.713
SMOTE - kNN	0.40	0.370	0.688
SMOTE - SVM	0.46	0.829	0.697
SMOTE - Random Forest	0.47	0.762	0.712
SMOTE - AdaBoost	0.49	0.784	0.726
RUS - AdaBoost	0.49	0.784	0.726
RUS - Boost	0.40	0.876	0.621
SMOTE - Boost	0.39	0.961	0.584

For problems with class imbalance, metrics such as precision, recall, and f1-score give good insight to how a classifier performs with respect to the minority class. Depending on the problem, the goal is to optimize precision and/or recall of the classifier. In this case, I want a model that catches the most number of instances of the minority class, even if it increases the number of false positives. A classifier with a high recall score will give the greatest number of potential customer churns, or at least raise a flag on most of the cases.

The resampled datasets all have higher recall percentages except kNN when compared to the original dataset. Three of the models have recall rates (for True) that are higher than 80% (Logistic Regression, SVM, RUS Boost) while one has a recall rate of 96%

(SMOTE Boost). The trade off is the accuracy and precision which means the model will have false positives. However, since the overall goal is to predict which customers might churn and keep them as a customer it is better to have false positives rather than miss customers that churn.

Conclusions and Future Work

The initial hypothesis was that customer churn could be predicted based on past customers. After doing exploratory data analysis, applying inferential statistics and extended analysis the hypothesis cannot be rejected. It appears that customer churn can be predicted based on seven dataset features after the data is resampled to increase the minor class.

Future Work

Additional work can be done to further improve the results. I'd suggest the following areas of study:

1. Gather more data
2. Rerun the various models with differing degrees of the 7 features
3. Apply multiple models to create ensembles
4. Apply anomaly detection algorithms to detect churn

Recommendations for the Client

With the initial analysis complete, I'd recommend gathering further data to conduct additional testing on the various models to understand the top two. Once that is determined, I'd recommend placing the top two models into production. I'd recommend splitting the data into three groups:

1. Model #1
2. Model #2
3. Control

By having three groups the data can be further analyzed to determine which group is performing best compared to the third group which is the control group. The control group is used to understand what would happen if no model had been deployed and it allows the data to be used as a comparison.

As testing proceeds, I would recommend adjusting the models to improve the performance based on recall. There might also be subsets of data that can further be explored once churn is realized with the models.

Resources / Resources Used

The following is a list of various resources used in this study.

1. https://www.kaggle.com/blastchar/telco-customer-churn
2. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
3. https://medium.com/greyatom/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b
4. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html#sklearn.metrics.classification_report
5. https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
6. https://imbalanced-learn.org/en/stable/generated/imblearn.over_sampling.SMOTE.html#r001eabbe5dd7-1
7. https://www.svds.com/learning-imbalanced-classes/
8. https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74
9. https://www3.nd.edu/~dial/publications/hoens2013imbalanced.pdf
10. https://sci2s.ugr.es/keel/pdf/algorithm/congreso/kubat97addressing.pdf
11. https://curate.nd.edu/downloads/0p096684s7g
12. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
13. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html
14. https://scikit-learn.org/stable/modules/tree.html
15. https://scikit-learn.org/stable/modules/neighbors.html
16. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
17. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
18. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
19. https://github.com/dialnd/imbalanced-algorithms
20. https://github.com/foutaise/texttable/