Abstract

Churn describes the loss of customers and is an important metric for telecommunications companies as it is easier and more profitable to retain customers than acquire new ones. Our main result is to create a simple yet useful model to explain the most important features that affect customer churn. We create two simple classification models of Logistic Regression and Decision Tree and compare their performance on our unbalanced dataset. The best overall model is used to explain the crucial features affecting Churn as well retention of customers.

Introduction

We will apply two data analytic models of Decision Tree and Logistic Regression on telecom data to predict whether or not a customer will leave Telco, a fictional telecom company. Predicting *customer churn* (also known as customer attrition) is an important piece of information across all industries because businesses are not only concerned with attracting new customers but also with retaining current customers. If a business can better understand what factors may cause a person to be more likely to leave, then the business can adjust its marketing and corporate strategy in hopes of lowering their customer churn. The telecommunications industry in particular is especially concerned with decreasing customer churn since, in most countries, a customer can choose from several different companies and the process of switching telecom providers is fairly easy. The ability to accurately predict which customers are more likely to churn can lead to significant monetary benefits for a company since, according to the *Harvard Business Review*, "acquiring a new customer is anywhere from 5 to 25 times more expensive than retaining an existing one," which means reducing customer churn is a significantly less costly option than increasing the number of new customers (Gallo). The challenge of such a study is to find a simple yet accurate model to explain why customers decide to Churn or not churn. Overcoming this challenge would help Telco understand why customer Churn and what steps they can take to avoid in an easy to understand manner.

Relevant Studies

Applying statistical analysis techniques to better predict customer attrition is not only a theoretical exercise, but is regularly conducted by various businesses across the globe. In the study we looked at, the authors also analyzed the Telecom dataset. Two important assumptions made: each customer that left the company would cost $500 to replace and every customer could be retained by spending $100 on them. To measure their success, the authors compared their model's performance against a dummy model that spends $100 per customer. They used a stratified train-test-split to split their data into a training set and test set. In addition, in order to prevent their model from under-classifying their target variable, they used SMOTE from imblearn.over_sampling so that the minority class would be 50% of their dataset. After performing train-test-split, the authors plotted a ROC curve. They optimized the four models (Logistic Regression, Gradient Boosting, Random Forest, and AdaBoost) based on recall as their

target metric. Through comparing the recall percentage and net amount of money saved amongst models, the authors determined that their Logistic Regression model performed best with a recall of 81% and net savings of $272,000 compared to the dummy model (Heintz).

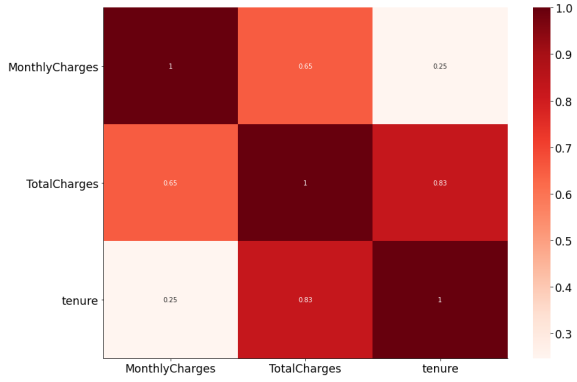
Problem Statement: Our report aims to find a *simple* model that can *easily and logically explain* why the customer decides to Churn or Retain based on the given dataset.

Data Exploration

Our dataset contains customer information for a fictional telecom company, Telco. While we found our dataset on Kaggle, the dataset was originally put forth by IBM as the data module *Telco Customer Churn* (Telco). We are interested in predicting the churn value (either "Yes" or "No") for a particular customer. The churn value indicates whether a particular customer has left the company within the last month (resulting in a "Yes") or remains a customer at the company (resulting in a "No"). The raw data contains 7,043 rows with one row per customer and 21 features (explained in more detail below). Eleven of these rows had NA value for TotalCharges feature and hence were removed from the dataset.

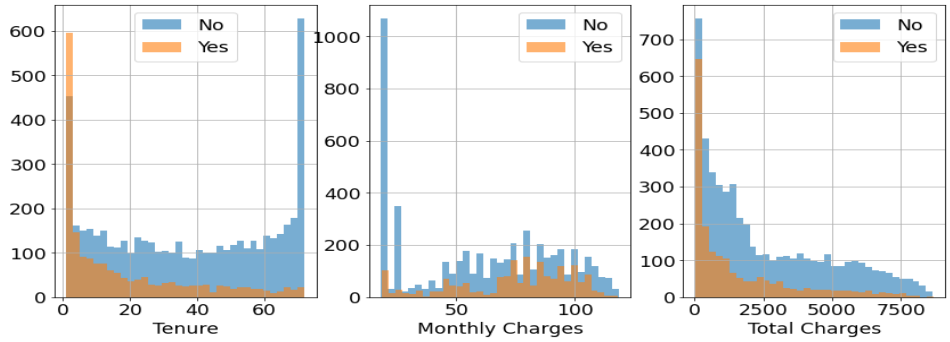
Numeric Features

Figure 1: Correlations between numeric features



There are three numeric features of this dataset - Tenure (the number of months the person has been a customer), MonthlyCharges and TotalCharges. Multicollinearity, which occurs when two or more features are shown to be highly related, can lead to calculations involving an individual feature to be inaccurate, but it does not reduce predictive power and reliability of the model as a whole at least within the sample set. As seen in Figure 1, there appears to be a high correlation (0.83) between TotalCharges and tenure, a moderately high correlation between TotalCharges and Monthly charges (0.65), and a moderately low correlation between MonthlyCharges and tenure (0.25). We convert Tenure to categorical based on Years to have a better understanding of the effect of Tenure on Churn.

Figure 2: Relationship between Churn and Numeric Features



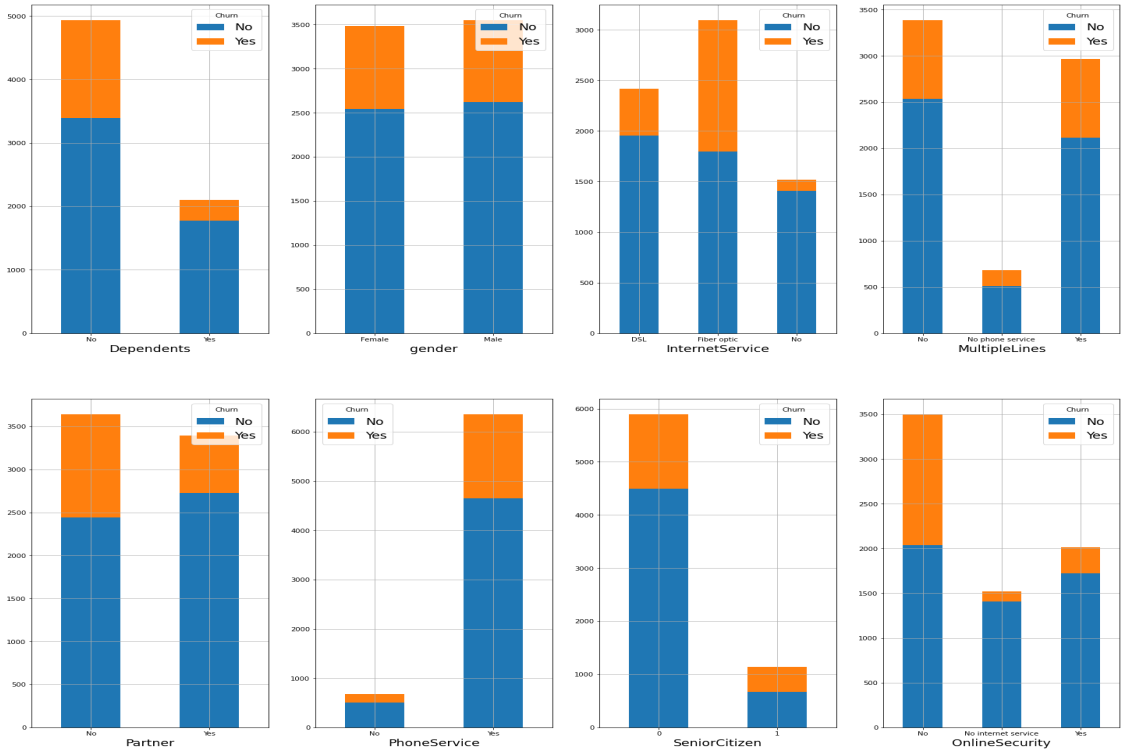
As observed in Figure 2, an extremely high number of customers have monthly charges less than $25. The distributions of churn values for customers with monthly charges greater than $30 are quite similar. The distribution of total charges is positively skewed regardless if the customer churned or not. The distributions for tenure are markedly different between customers who churned and those who did not. Figure 2 shows that the distribution is positively skewed for customers who churned and negatively skewed for customers who have not churned. This leads us to conclude that customers who churn likely cancel the service in the first few months of joining the company. There appear to be two spikes, with the second spike denoting a large group of customers who have been using the service for more than five years and are more prominent than the first spike.

Categorical Features

There are nineteen categorical features of this dataset - gender (male or female), Senior Citizen (1 for Yes, 0 for No), Partners (Yes or No), Dependants (Yes or No), PhoneService (Yes or No), MultipleLines (Yes, No and No phone Service), Contract (Month-to-Month, One Year, Two Year),PaperBiling (Yes or No), Payment Method (Bank Transfer, Credit Card, Electronic and Mail Check) and OnlineSecurity, OnlineBackUp, Device Protection, TechSupport, StreamingTV, StreamingMovies (all having Yes, No or No Internet Service as an input)

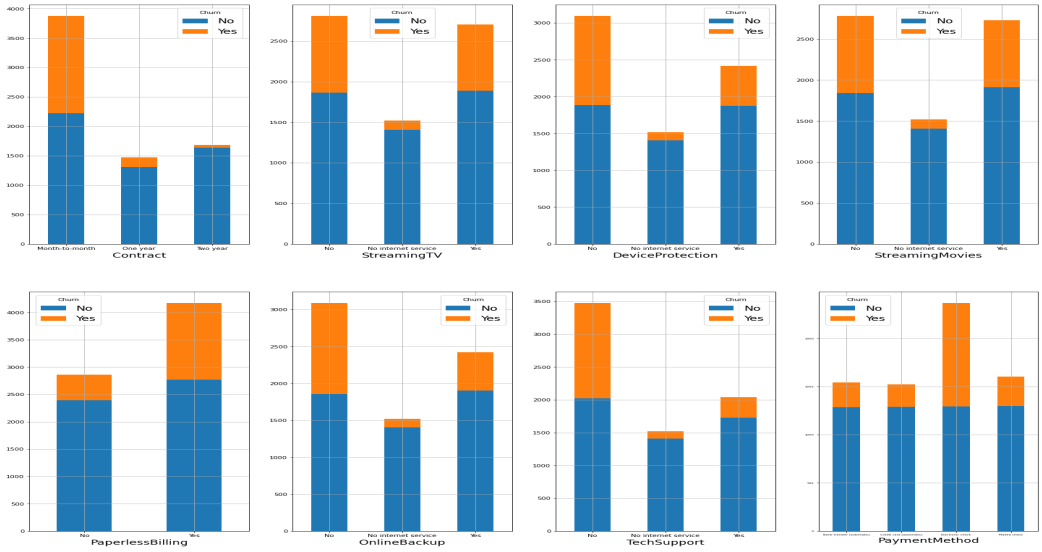
Relationship between Churn and categorical features

Figure 3: From left to right: (a) Dependents, (b) gender, (c) InternetService, (d) MultipleLines, (e) Partner, (f) PhoneService, (g) SeniorCitizen, (h) Online Security



If the value of PhoneService is "No," then the value of MultipleLines is "No phone service." This means that the "No phone service" column in Figure 3d does not actually hold any predictive power. Similarly, if the value of InternetService = "No," then the value of OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTv, and StreamingMovies all had a value of "No internet service." Thus, the "No service column" for those features does not have any predictive power.

Figure 4: from left to right (a) Contract, (b) StreamingTV, (c) DeviceProtection, (d) StreamingMovies, (e) PaperlessBilling, (f) OnlineBackup, (g) Techsupport, (h) PaymentMethod



As seen in Figure 4f, 4c, and 4g, customers who opted into these services (OnlineSecurity, DeviceProtection, OnlineBackup, and TechSupport, respectively) have a lower churn rate than customers who did not opt into these services. In contrast, Figure 3d, Figure 4b and 4d do not show a large difference in churn rates with customers who do or do not have these services. Another observation we noticed is that an increase in contract length led to a decrease in churn rate as seen in Figure 4a. Additionally, the ways in which customers are billed and pay for Telco's services seem to have an impact on churn rates. In Figure 4e, customers with paperless billing have a higher churn rate than those without. In Figure 4h, customers who pay with electronic check have a higher churn rate than those who pay using alternative methods. However, one should note that for some features, the churn rate does not significantly differ. In Figure 3b and 3f, the churn rate remains fairly consistent for both response values. Demographics also seem to have some effect on churn rates. One may also note that senior citizens (value > 0.5) , as shown in Figure 3b, have a higher churn rate than those who are not senior citizens (value < 0.5). Customers with partners or dependents (Figure 3e and 3a, respectively) have a lower churn rate than customers who are single or do not have dependents.

Methodology

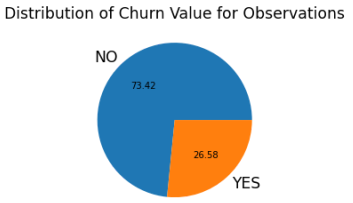
Please note that if one wishes to replicate our work, scikit-learn version 0.22.2.post1 and Python 3.6.9 was used.

We performed the remainder of our statistical analysis on a dataset of 7,032 customer records (excluding the 11 observations for which TotalCharges were empty). In addition, we removed the customerID column from all entries as a customer's ID was only present for unique identification and therefore would not be relevant in training the model.

After removing the feature and data entries as detailed above, we converted Churn (target) into 0 for No and 1 for Yes. We used train_test_split to divide 70% of the data into a training set and the remaining 30% into a test set with random state set to zero to allow for repeatability when creating these random testing and training sets.

After constructing our training and test datasets, we transformed numerical and category features respectively. Standard scaling is used on numerical features to rescale the features so they have the standard distribution (mean of zero and standard deviation of one) to reduce dominance of features with big ranges. Standard Scaling has the biggest effect on Logistic Regression because Regularization (eg. L1 or L2) makes input features dependent on the scale of the feature (matthiash). One Hot Encoding is used on the categorical features converted into a form that could be provided to ML algorithms without natural ordering between categories such as Male is 0 and Female is 1 so that it cannot assume Female > Male. For each category, a vector of the same length as all unique values of class is added with binary values where 1 is at a specific position (Brownlee).

In order to ensure our model was not only accurate but also easy to explain - two of the most famous and easy classification models are used - Logistic Regression and Decision Tree.



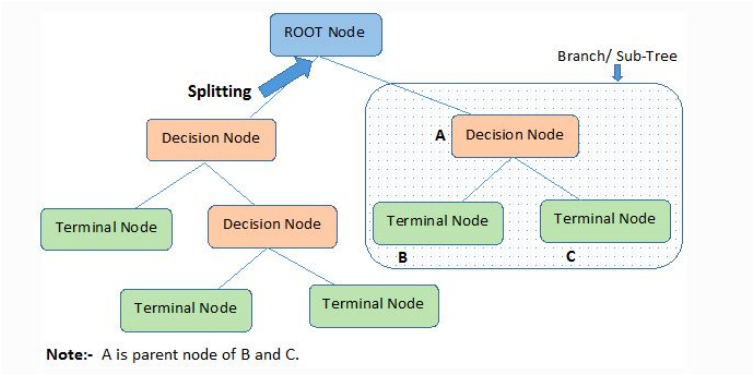
K-Nearest Neighbour and Naive Bayes were not used due to limited visualisation because of high number of input features and correlation between features respectively as talked above.

A basic structure is taken to fitting and testing the models. Using pipeline, all the transformation is done on all the input features before choosing the model. Most model have random state initialized to zero to ensure same metric values for repeatability of the report and class weight to balanced uses the values of Churn (0 or 1) to automatically adjust weights inversely proportional to class frequencies in the input data - giving more weightage to the minority class (Wang). As seen in the figure on the right, our dataset's minority class is Yes with only 26.58% observations.

Using GridSearchCV, exhaustive combinations of all the parameter grid features are used to hyper tune the model and retain the model with best parameters based on the metric of macro f1-score - used due to giving a better understanding of the model with unbalanced classes and also allows for giving equal importance to Yes or No for Churn (Scikit). For training our model to find the best parameters, GridSearchCV uses 5-Stratified Cross validation. While cross validation allows to train and improve the model on the training set whereas stratification seeks to ensure that each fold is representative of all strata of the data by equally represented in the validation set to improve the model. (Amelio Vazquez-Reina) In order to perform cross validation, we first ensured the dataset had been randomly shuffled. We then split the dataset into k = 5 folds, and for each iteration, we selected one fold (that had not been selected previously) to be the test dataset and used the remaining k-1 folds to train our model. We recorded the accuracy achieved on each iteration and after performing k iterations, set the model's performance score equal to the average accuracy of the k iterations (Sanjay.M).

Finally, confusion matrix as well as other important metrics such as weighted f1-score, accuracy, precision and recall are calculated for testing and training set to understand and compare the model's performance. Macro F1-score, is the harmonic mean of true positive rate and precision, and is a better indicator of the predictive ability of the models given the imbalance in target classes by giving equal weightage to majority and minority class (Tharwat)

Decision Tree Algorithm



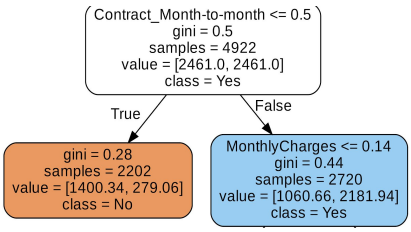
Decision Tree algorithm is a supervised learning algorithm. The main aim of this algorithm is to predict the class or the target feature by learning simple decision rules inferred from prior data (training data). As the target data is Categorical (0 or 1), the

type of tree used is a Categorical Variable Decision Tree. A basic structure of a decision tree has root nodes, decision nodes and leaves nodes as seen in the diagram below.

Nodes are split based on providing the highest information gain for a given criteria (our model uses gini Index as the criteria). Information gain can be loosely defined as the difference between Information before and after the parent node split (KDnuggets). Gini Index (I_G) is the number (p_j) of samples (c) that belong to a specific class for a particular node.

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

A sample calculation of information gain based on the Gini criteria is as follow for the given example below:



Gini for Parent Node: $1 - ((\frac{2461}{4922})^2 + (\frac{2461}{4922})^2) = 0.5$

Gini for Left Child Node: $1 - ((\frac{1400.34}{2202})^2 + (\frac{279.06}{2202})^2) = 0.28$

Gini for Right Child Node: $1 - ((\frac{1060.66}{2720})^2 + (\frac{2181.94}{2720})^2) = 0.44$

Information Gain: $0.5 - 0.44(\frac{2720}{4922}) - 0.28(\frac{2202}{4922}) = 0.1316$

The aim is to get Gini to zero for the leaf nodes so they are pure leading to no more information gain. Feature importance is the total amount that the gini index decreases due to splits over a given feature and the sum of all feature importance is 1. It gives an idea on which features are important for trees based on Gini Criteria.

Important Parameter Grid Features

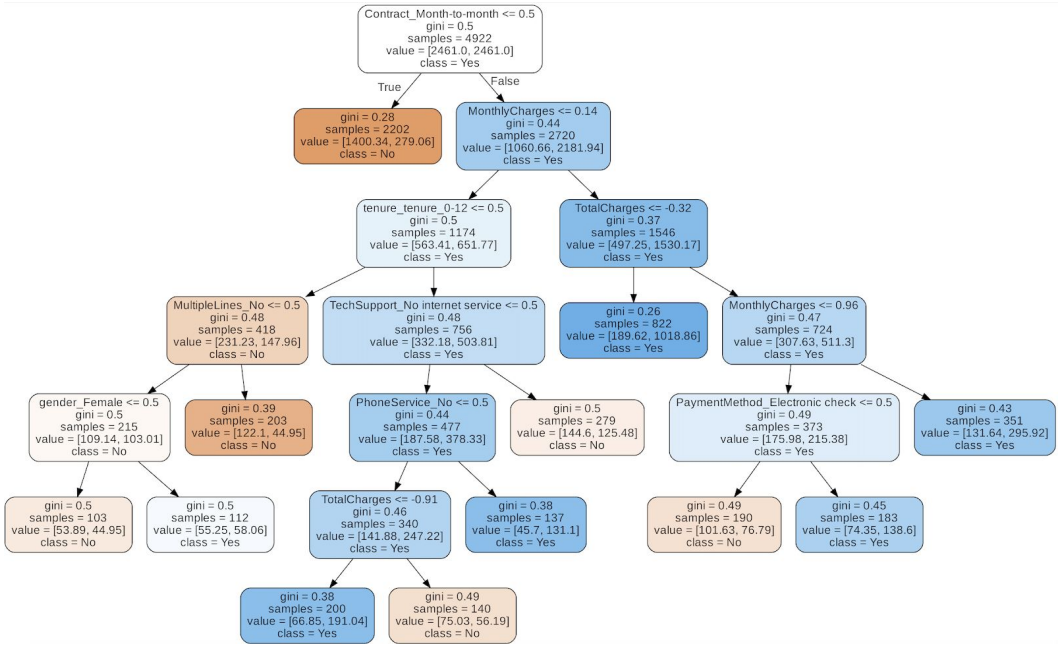
The following parameters of Decision Tree when hypertuned with GridSearchCV did not go to default values and hence are important:

- *Max Features:* The number of features to consider when looking for the best split. Otherwise, uses all the features. (Ceballos)

- *Minimum Samples Leaf:* This is used to limit the growth of the tree by setting the minimum number of samples needed to be considered a leaf node. (Ceballos)

Visualization

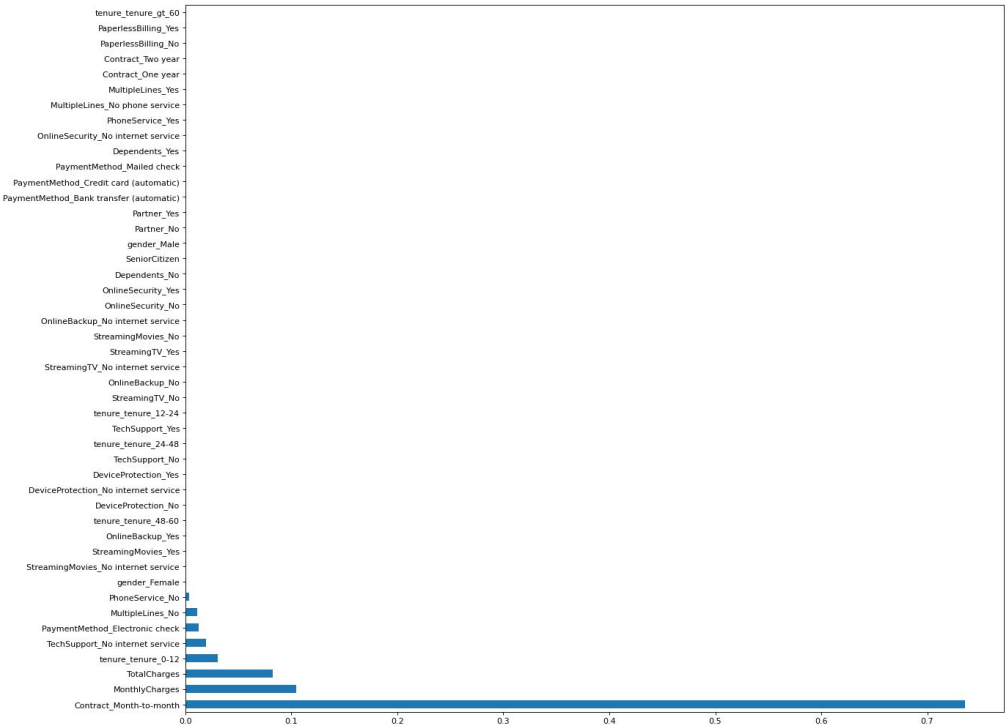
Before visualising the tree, it was post-pruned (pruning is supported by sklearn) to remove all the children of a parent node if all children nodes have the same classification.



In order to read the decision tree:

- Left Child means the condition is True and vice versa.
- For categorical features, less than and equal to 0.5 means 0 (No) and vice versa.
- Numerical features are standardized and hence they can be from 0 to 1.
- Class refers to whether the leaf is going towards the Churn value of Yes or No.

We also plot the feature importance:



Based on this graph, five most important features are Contract, Monthly Charges, Total Charges, Tenure and Tech Support. Hence, these features are very important for Decision Tree to split the observation into Churn Yes or No.

Performance

	Accuracy	Macro F1-score	Macro Precision	Macro Recall
Train	0.7641	0.7276	0.7175	0.7581
Test	0.7550	0.7151	0.7055	0.7451

Logistic Regression Algorithm

Another model we used to predict churn is Logistic Regression. In the Logistic Regression model, the dependent variable assumes two values: zero or one. Thus, this model is used for predictions in which there are only two possible outcomes, which makes the model suitable for our task of predicting whether a customer has churned. Logistic Regression uses the logistic function (with the features we are using):

$$p(X) = \frac{e^{\beta 0 + \beta 1X1 + \beta 2X2 + \ldots + \beta nXn}}{1 + e^{\beta 0 + \beta 1X3 + \beta 2X2 + \ldots + \beta nXn}}$$

In the logistic function, p(X) is the probability of the particular data X and βj refers to the jth feature. We must estimate the coefficients βj for our j =1,2...n features. With this function, one should note that the result will be a number between 1 and 0; every time we obtain a feature greater than 0.5 we classify it as 1 and if it is lower we classify it as 0 (Gareth et al., 2015).

Our Logistic Regression model uses L2 Regularization. Regularization allows the model to avoid over-fitting by penalizing high coefficient features (stop growing boundlessly). The Least Square Error that helps in fitting the model by decreasing it as much as possible as its the error between predicted and true value however, it is unstable especially with multicollinearity (present in our dataset with MonthlyCharge and TotalCharge) (Stephanie). L2 Regularization adds the square of magnitude of coefficient to LSE and is good at dealing with correlated inputs.

Important Parameter Grid Features

The following parameters of Logistic Regression when hypertuned with GridSearchCV did not go to default values and hence are important:

- *C*: It is the inverse of regularization strength. When there are too many features but few observations, the model has a tendency to overfit. To solve this issue as well as reduce

error, multiply lambda (regularization strength) to the square of magnitude of coefficient (L2) and hence, C is the inverse of the lambda. (user3427495)

- *Solver*: It tries to find the feature weights that minimize the cost function. (Hale)

Visualization

Based on the coefficients, the four most important features that push the model towards Churn value of Yes are Monthly Charges (0.87), Tenure of less than year (0.83) , Monthly Contract (0.79) and Electronic Check Payment Method (0.32). Similarly, the four most important features that push the model towards Churn value of No are Two Years Contract (-0.78), Tenure of greater than 60 months (-0.38) and having TechSupport (-0.24).

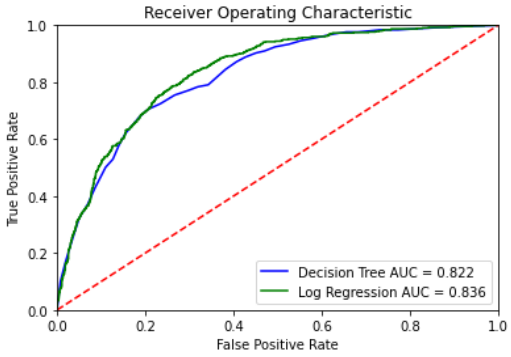
Performance

	Accuracy	**Macro F1-score**	**Macro Precision**	**Macro Recall**
Train	0.7495	0.7350	0.7199	0.7640
Test	0.7517	0.7189	0.7114	0.7608

Comparison

In this section we compare the models. The classification report can be seen in the following table.

Model	Accuracy (Test)	Recall	Precision	F1-score	Area under curve (AUC)
Decision Tree	0.755	0.7451	0.7055	0.7151	0.8222
Logistic Regression	0.7517	0.7608	0.7114	0.7189	0.8357



The above table suggests that all models test comparable with Logistic Regression edging them out in performance by around 0.4. It is important to note that null accuracy for the test set is 73.70% and Logistic Regression performs worse in accuracy compared to Decision Tree.

Another measure that we employed is the Receiver Operating Characteristics (ROC) curve for all of the models. The ROC curve is obtained by changing the threshold value and evaluating the combination of false

positive rate and true positive rate for each threshold. While evaluating the models, we set the threshold as 0.5. This threshold can be interpreted as the value in which the model will decide if the Churn is equal to one or zero. If the probability returned by the model is higher than the threshold, the model assumes churn equals 1.

One measure that we can use from the ROC curve is the area under the curve (AUC). While the ROC compares different thresholds, the AUC gives us an average of performance for various thresholds. It is possible to see that, although the Logistic Regression has an AUC of 0.836 higher than Decision Tree for the threshold 0.5, when we consider the average performance for different thresholds, Logistic Regression does a better job of predicting Churn status based on the ROC curve.

We find Logistic Regression to be the best model with its high Macro f1-score, which serves as a proxy for the actual performance of the classifier, as well as a high AUC value, which tells how well the two kinds of responses can be separated (Normanius). As our data contains more overlapping regimes then Logistic Regression should perform better as this leads to overfitting of Decision Tree also known as data snooping bias and hence, explains Logistic Regression's superior performance Gareth et al., 2015).

The simplicity of Logistic Regression is also worth nothing. Thus, we can use the coefficient values to understand what factors may lead to someone Churning on our Telecom Company.

Features Value	Churn	Reason
Monthly Charges	*Yes*	Price is the biggest reason to switch and higher prices pushes customers to find competitors with cheapest service.
Tenure (0-12 Months)		Customers are learning to use your product and deciding whether or not to stay with it
Contract (Month to Month)		12 purchasing decisions per year and hence, more time to reflect on the cost-effectiveness of Telco's services
Contract (Two Years)	*No*	Customers have only one purchasing decision per two year (albeit a larger one)
Tenure (More than 60M)		Customers will have enough time to implement the product and see the benefits of using it and hence, are more likely to commit.
Total Charges		The more customers have spent on the company, the more likely they are to not Churn.

Conclusion

Customer retention is of great importance to telecom companies to determine effective business policies and remain competitive in the market. By analyzing the factors affecting customer churn, our project can help telecom companies understand their customers' needs, provide better services and generate greater profits in the long run.

After initial exploratory analysis, we transformed the column before including the features in our predictive models.We then trained models of Decision Tree and Logistic and fine tuned their hyperparameters to get us the best train macro F1-score. It is important to note that each model has its pros and cons and that which model performs best depends on the intrinsic relationship between the response and features.

Our results show that Logistic Regression outperforms Decision Tree with 0.7189 Macro F1-Score as well as 0.155 higher AUC value - meaning that it is overall better at distinguishing Churning and Non-Churning Customers even as a simple model. We also learned that Contract length, Tenure and Charges plays a proportional role with Churning.

One of the Limitation of our data is multicollinearity in not only numerical features with Monthly and Total Charges but also in categorical features like Internet Service - A value of 'No' for internet service would automatically mean that all online services would have a value of 'No internet service'. For this, Principal Component Analysis (PCA) is a technique for feature extraction in which existing features are combined together to create new input features where least important features are dropped. This not only allows for independence between the new features but also allows for dimensionality reduction (Upreti). Hence, even though it may make the input features a bit harder to explain, it will also allow for more robust models of Naive Bayes and K-Nearest Neighbours respectively.

References

1. Amelio Vazquez-Reina. "Understanding Stratified Cross-Validation." *Cross Validated*, 1 Nov. 1962, stats.stackexchange.com/questions/49540/understanding-stratified-cross-validation.
2. Brownlee, Jason. "Why One-Hot Encode Data in Machine Learning?" *Machine Learning Mastery*, 19 May 2019, machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.
3. Ceballos, Frank. "Scikit-Learn Decision Trees Explained." *Medium*, Towards Data Science, 6 Apr. 2020, towardsdatascience.com/scikit-learn-decision-trees-explained-803f3812290d.
4. Gallo, Amy. (2014). The Value of Keeping the Right Customers. *Harvard Business Review.* Retrieved from https://hbr.org/2014/10/the-value-of-keeping-the-right-customers
5. Gareth, J., Witten, D., Hastie, T., Tibshiranin, R. (2015). *An introduction to statistical learning: with Applications in R (6th ed)*. New York, NY: Springer.
6. Hale, Jeff. "Don't Sweat the Solver Stuff." *Medium*, Towards Data Science, 7 Apr. 2020, towardsdatascience.com/dont-sweat-the-solver-stuff-aea7cddc3451.
7. Heintz, Brenner. (2018). Cutting the cord: Predicting customer churn for a telecom company. Retrieved from https://towardsdatascience.com/cutting-the-cord-predicting-customer-churn-for-a-telecom-company-268e65f177a5
8. "Understanding Decision Trees for Classification in Python." *KDnuggets*, www.kdnuggets.com/2019/08/understanding-decision-trees-classification-python.html.
9. matthiash, and Ami Tavory. "Logistic Regression and Scaling of Features." *Cross Validated*, 11 July 2017, stats.stackexchange.com/questions/290958/logistic-regression-and-scaling-of-features.
10. Normanius (2019, Sep 4). Reason of having high AUC and low accuracy in a balanced dataset. [Msg 2]. Message posted to https://stackoverflow.com/questions/38387913/reason-of-having-high-auc-and-low-accuracy-in-a-balanced-dataset
11. Sanjay.M. (2018). Why and how to cross validate a model. Retrieved from https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f

12. *Scikit* "3.2. Tuning the Hyper-Parameters of an Estimator¶." *Scikit*, scikit-learn.org/stable/modules/grid_search.html.
13. Stephanie. "Regularization: Simple Definition, L1 & L2 Penalties." *Statistics How To*, 14 Oct. 2018, www.statisticshowto.com/regularization/.
14. Telco Customer Churn. (2017). *Kaggle* [Data file]. Retrieved from https://www.kaggle.com/pavanraj159/telecom-customer-churn-prediction/data
15. Tharwat, Alaa. (2018). Classification assessment methods. *Applied Computing and Informatics.* https://doi.org/10.1016/j.aci.2018.08.003
16. Upreti, Jainendra. "Principal Component Analysis - Case Study." *Https://Rstudio-Pubs-Static.s3.Amazonaws.com/*, 16 July 2017, rstudio-pubs-static.s3.amazonaws.com/291945_5facb930925d4fcdbc9b53d2a0ec6f53.html.
17. user3427495, and TooTone. "What Is the Inverse of Regularization Strength in Logistic Regression? How Should It Affect My Code?" *Stack Overflow*, stackoverflow.com/questions/22851316/what-is-the-inverse-of-regularization-strength-in-logistic-regression-how-shoul.
18. Wang, Zichen. "Practical Tips for Class Imbalance in Binary Classification." *Medium*, Towards Data Science, 24 Jan. 2019, towardsdatascience.com/practical-tips-for-class-imbalance-in-binary-classification-6ee29bcdb8a7.