A0168163B BT2101

Analysis of Customer Churn Data and Prediction of Churn

Data clean up

The Data that was provided is surprisingly clean. There are a few cells in the 'TotalCharges' column that were empty. We notice that this is due to the tenure being 0 and that the customer has yet to pay the first month's bills. Hence, the empty cells should logically be "0" and the data is not "missing". The categorical data was also binarized (refer to notebook).

Data Pre-Processing and Analysis

The feature of interest that we wish to determine would be the "Churn" column. We notice that 26.5% of the entries have a "Churn" that is "Yes" and 73.5% a "No" [1]. We can also get a statistical summary of the data columns that are numerical [2]. The data is unbalanced, and there is a lot more customers with "Churn" that is "No" as compared to "Yes". We will need to balance the data so that our model is not biased, since many classifiers are "sensitive to the proportions of the different classes", and also because the "value" of finding the minority class (i.e. Churn = "Yes") is higher than finding the majority class (i.e. Churn = "No"), since Telcos will be more interested in the customers that are likely to Churn as compared to customers that are not likely to Churn. This is an important point that will guide how I select and fine tune my models later.

There is the option of discretizing the continuous variables, but I decided not to do so for mainly two reasons: 1) we have no knowledge on the scale of aggregation (we don't know how many bins to select) and 2) to prevent loss of information. The explanation in detail is in the notebook.

Cross Validation (Preventing Overfitting and Leakage of Data)

To prevent overfitting and to make sure we chose the right model, we first did a Stratified Split of the data (70 / 30 split), applied Synthetic Minority Over-Sampling (SMOTE) on the training set, did a K-Fold (K = 10) Cross Validation on several baseline models before deciding on one. We did a Stratified Split to prevent the (unlikely) event that a large majority of one classification ends up in either the Training or Test Set, since the proportion of Churn "Yes" is very low (26.5%) as compared to Churn "No" (73.5%). This method also ensures that there is no data leakage as we only perform these modifications on the Training Set to make sure we create an unbiased model with little overfitting. The Test Set was left untouched until we have finalized a model to prevent tainting the Test set.

Model Results

We chose a baseline model of a Decision Tree Classifier (Entropy) [3]. We supplemented by using the ensemble learning method of Gradient Boosting [4] Subsequently, we further fine tuned our model to increase the "recall" score. The reason for focusing on the "recall" score is that a telco company would be more interested in being able to identify customers who Churn (i.e. have high True Positive and Low False Negatives), therefore, a model is more useful if it can better predict a positive Churn (Churn = "Yes") as compared to a model that is less likely to label a "No" as a "Yes" (i.e. predicts “No” more accurately), since Telcos want to be able to better anticipate customer churn.

Therefore, we improved our Gradient Boosting model to increase Recall score at the cost of lower accuracy. The final scores can be seen here [5]. Gradient Boosting in general does suffer from potential Overfitting and hence, we also reduced the learning rate of our model to lower the possibility of Overfitting.

Notable/Important features

It seems that the more important features in determining Churn, are Contract (Month to month), Online Security, Payment Method (Electronic check), Tech Support, Online Backup and Tenure.[6]

If we were to intuitively guess from the start what are some features that may be of importance, most of us would have guessed Tenure as the most important factor (long withstanding customers = less likely to Churn?) but consider the Contract type (Month to month). A customer who is on a contract that is renewed each month is logically more likely to Churn as compared to customers who’s contract type is One-year or Two-years, as they are free to decide if they wish to continue with this Telco or switch to a different one very often (monthly), as compared to One/Two-year long contracts. Hence, this Contract (Month to month) being the most influential feature in determining customer Churn is logical.

Limitations/Other considerations

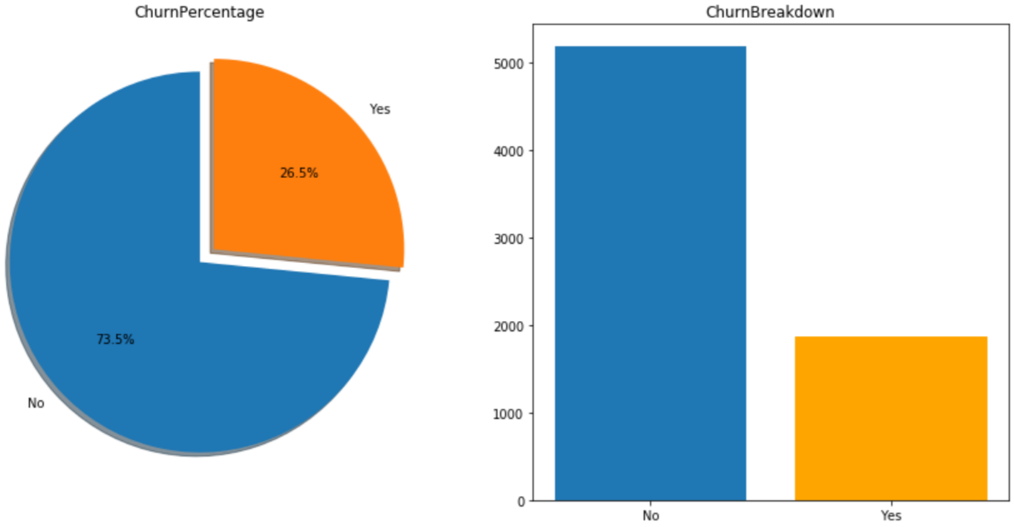
There is the problem of Overfitting since we did Synthetic Minority Over-Sampling. This creates “synthetic” data for the minority class, which is the when “Churn” is “Yes”. There is the possibility of a learning algorithm to generate a classification rule just to cover one single replicated example. Also, it “increases the number of training examples, thus increasing the learning time”. However, even after using SMOTE, it seems that Overfitting was not inherent in our model as our test data showed rather high scores when tested on our test data.

I would also prefer if did not have to generate Synthetic Data and have a balanced data set from the start, which would remove the need to do Over-Sampling and will produce a model with higher scores.

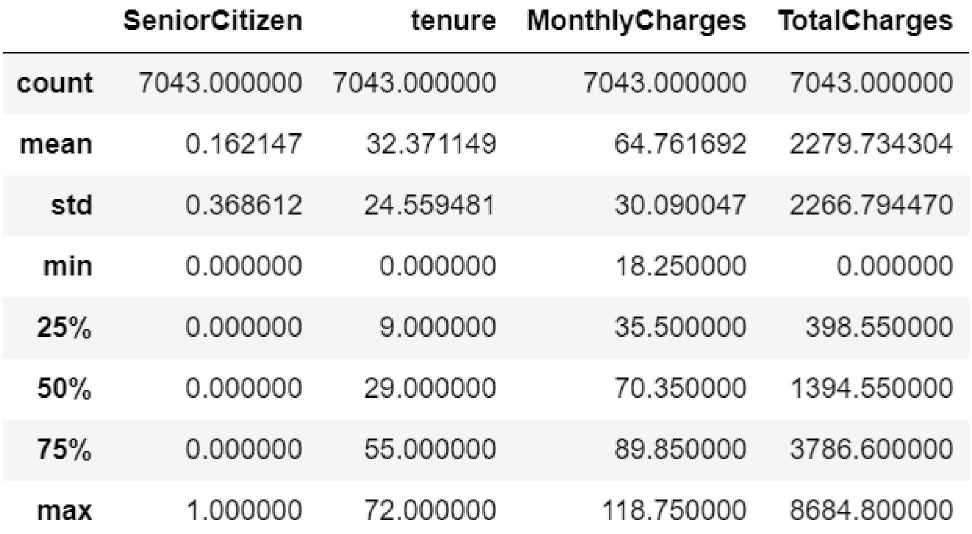
Our Gradient Boosting model, though has a decent accuracy of 76%, still poses certain limitations due possible lack of data. Firstly, your average Telco company will definitely have more than 7043 customers, hence we cannot confidently apply this model to a population. We can infer that we have been given a small sample of data to work with. Telcos like Singtel have an approximate 3.58 million subscribers for its mobile customer base and 7043 customers would barely amount to 0.2% of their customers. Our model is very restricted in that the sample data provided to us is extremely small.

Some additional data that would also have been useful is the different costs for the various “plans” a customer subscribes to from this Telco, which will allow us to do discretization for the “monthly” charges feature.

[1]

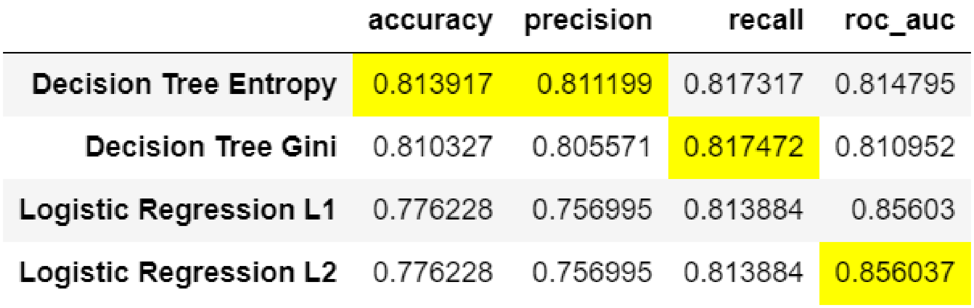


[2]



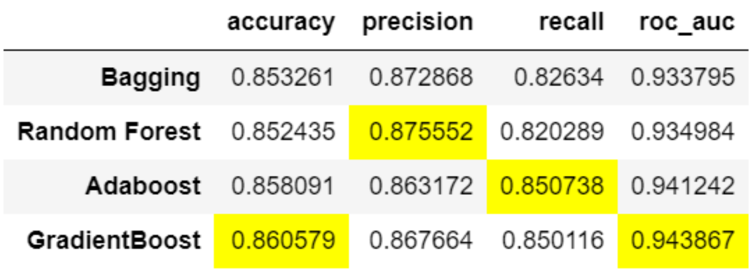
	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.734304
std	0.368612	24.559481	30.090047	2266.794470
min	0.000000	0.000000	18.250000	0.000000
25%	0.000000	9.000000	35.500000	398.550000
50%	0.000000	29.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

[3]



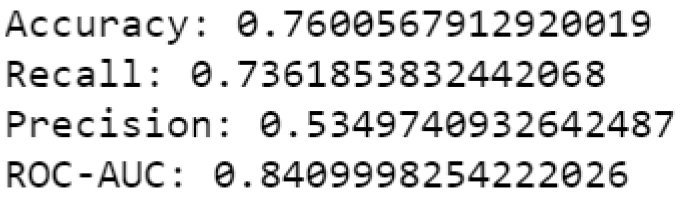
	accuracy	precision	recall	roc_auc
Decision Tree Entropy	0.813917	0.811199	0.817317	0.814795
Decision Tree Gini	0.810327	0.805571	0.817472	0.810952
Logistic Regression L1	0.776228	0.756995	0.813884	0.85603
Logistic Regression L2	0.776228	0.756995	0.813884	0.856037

[4]



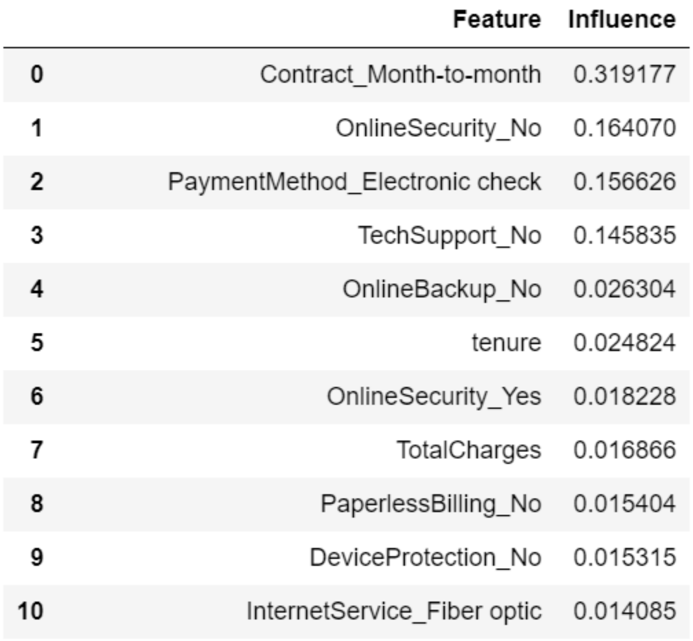
	accuracy	precision	recall	roc_auc
Bagging	0.853261	0.872868	0.82634	0.933795
Random Forest	0.852435	0.875552	0.820289	0.934984
Adaboost	0.858091	0.863172	0.850738	0.941242
GradientBoost	0.860579	0.867664	0.850116	0.943867

[5]



```
Accuracy: 0.7600567912920019
Recall: 0.7361853832442068
Precision: 0.5349740932642487
ROC-AUC: 0.8409998254222026
```

[6]



	Feature	Influence
0	Contract_Month-to-month	0.319177
1	OnlineSecurity_No	0.164070
2	PaymentMethod_Electronic check	0.156626
3	TechSupport_No	0.145835
4	OnlineBackup_No	0.026304
5	tenure	0.024824
6	OnlineSecurity_Yes	0.018228
7	TotalCharges	0.016866
8	PaperlessBilling_No	0.015404
9	DeviceProtection_No	0.015315
10	InternetService_Fiber optic	0.014085

References:

- Weiss, G.M., McCarthy, K., & Zabar, B. (2007). Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs? *DMIN*.

- Santacruz.A (Sep 20, 2016) Why it is important to work with a balanced classification dataset, Retrieved from: http://amsantac.co/blog/en/2016/09/20/balanced-image-classification-r.html

- Singtel (May 9, 2012) Singtel Group's mobile customer base reaching 445 million, Retrieved from: https://www.singtel.com/about-Us/news-releases/singtel-groups-mobile-customer-base-reaches-445-million