BANA 273 Machine Learning

Analysis on Telecom Customer Churn Rate

Team 19
Harinishri Srikanth, Pei-Ling, Wu, Praveen Aravindar, Xinfa Pan

Table of Content

1. Introduction

Executive Summary

In nowaday, people need to have their electronic devices such as phones, pads, and laptops around them to serve as productive tools and entertainment tools. Regardless of the purpose of people using electronic devices, the one thing people have in common is that people need the internet to function. Out of many companies that provide telecommunication such as internet and phone services, customers can easily switch to other companies for internet service. Therefore, it is important to understand what attributes cause customers to switch to other telecommunication companies.

Business Idea

Considering a Telecom company called Telco, many customers leave the company for various reasons. Due to this, the company may face a huge loss. The ultimate solution we are trying to find out is what attributes play an important role in determining the Churn rate. By implementing this analysis, we will be able to identify the key attributes and try to find out the major reason for the customers to leave the company.

This is a pre-crawled dataset, taken as a subset of a bigger dataset that was created by extracting data from IBM, which ranks among the world's largest information technology companies. We would like to investigate the relationship between the Churn rate and other attributes in the dataset such as Tenure, Phone service etc.. This is important for the company because our analysis should indicate the most significant and priority attributes that contribute to the Churn rate. Thus, it can help decide what kind of service to provide as a priority to attract more customers and finally achieve the target of reducing the Churn rate on the application. The Machine learning models that we will be using are Decision Tree, NaiveBayes and RandomForest in WEKA.

2. Data

We found "WA_Fn-UseC_-Telco-Customer-Churn.csv" on Kaggle, which contains 7043 rows (customers) and 21 columns (features).

Columns description:

	Column	**Description**	**Sample value**
1	customerID		7590-VHVEG
2	gender	Whether the customer is a male or a female	male or a female
3	SeniorCitizen	Whether the customer is a senior citizen or not	(1, 0)
4	Partner	Whether the customer has a partner or not	(Yes, No)
5	Dependents	Whether customer has dependents or not	(Yes, No)
6	tenure	Number of months the customer has stayed with the company	34
7	PhoneService	Whether the customer has a phone service or not	(Yes, No)
8	MultipleLines	Whether the customer has multiple lines or not	(Yes, No, No phone service)
9	InternetService	Customer's internet service provider	(DSL, Fiber optic, No)
10	OnlineSecurity	Whether the customer has online security or not	(Yes, No, No internet service)
11	OnlineBackup	Whether the customer has online backup or not	(Yes, No)
12	DeviceProtection	Whether the customer has device protection or not	(Yes, No)
13	TechSupport	Whether the customer has tech support or not	(Yes, No)
14	StreamingTV	Whether the customer has streaming TV or not	(Yes, No)

15	StreamingMovies	Whether the customer has streaming movies or not	(Yes, No)
16	Contract	The contract term of the customer	(Month-to-month, One year, Two year)
17	PaperlessBilling	Whether the customer has paperless billing or not	(Yes, No)
18	PaymentMethod	The customer's payment method	(Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic)
19	MonthlyCharges	The amount charged to the customer monthly	
20	TotalCharges	The total amount charged to the customer	
21	Churn	Whether the customer churned or not	(Yes, No)

Output:

Churn - Target class

3. Visualization

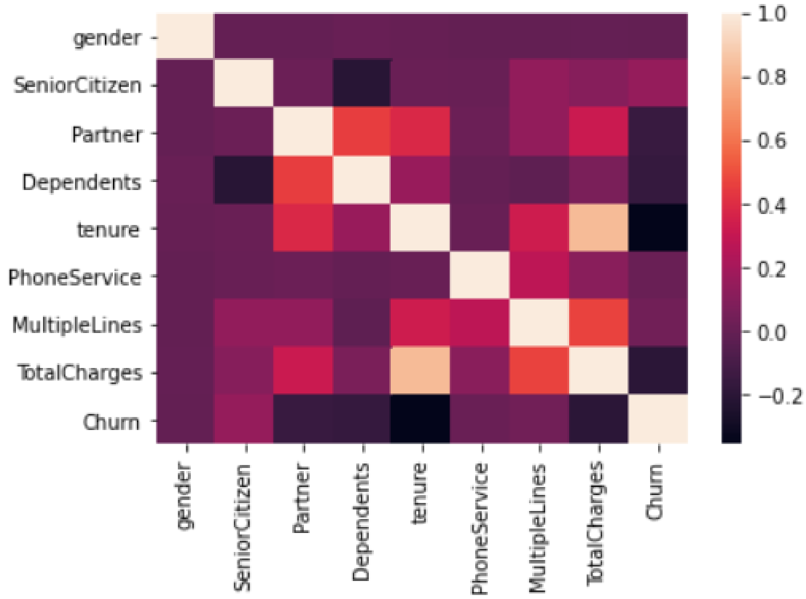

Figure 1 : HEATMAP

We used a heatmap to see some of the columns with binary values in the dataset to see the correlations between attributes, and most importantly correlations with Churn rate. It turns out that these attributes all have correlations under 0.3 between the Churn rate.
In contrast, Partner has high correlation with Dependents, tenure, and TotalCharges. Tenure has high correlations with total charges and multiple lines .

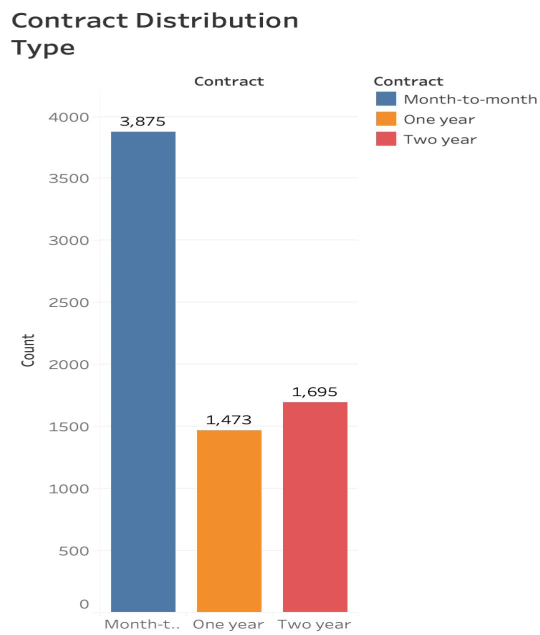

Figure 2 : Contract Distribution

We used a Contract Distribution bar graph to understand the number of customers using either Month-to-month or One Year or Two year contract. From this we come to know that a large number of customers have taken Month-to-month contracts, which is more than the sum of customers who have taken One Year or Two Year contracts.

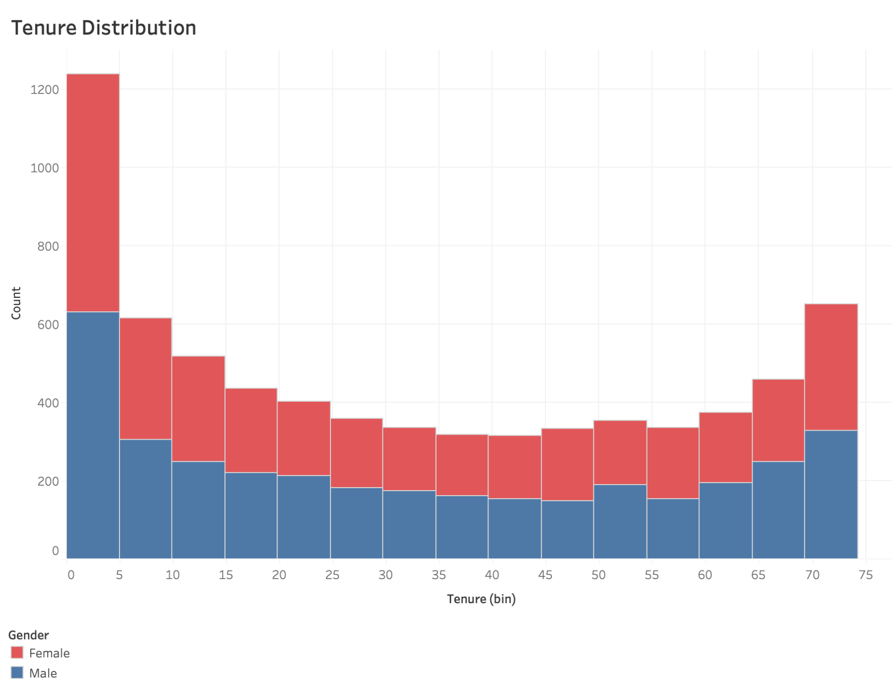

Figure 3 : Tenure Distribution

The Histogram depicts the Distribution of Tenure through bins and also clearly explains the difference among the number of male and females using the service.

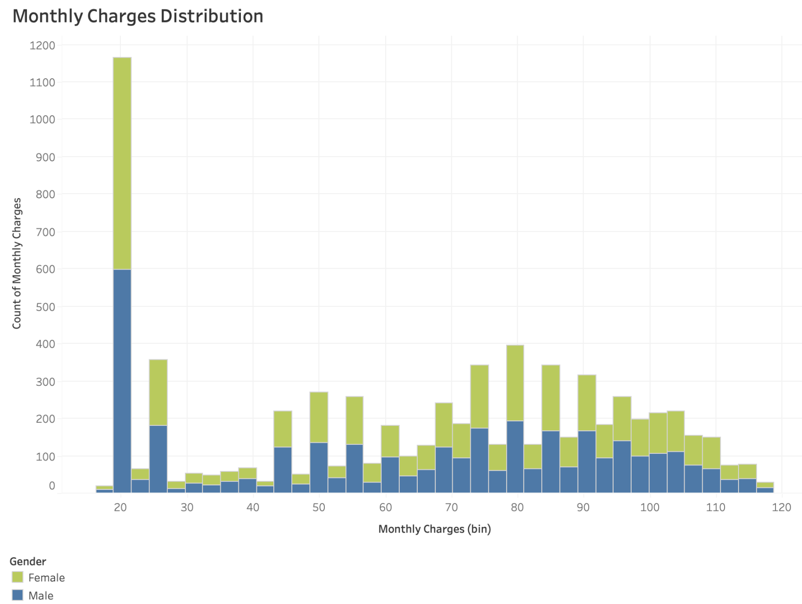

Figure 4 : Monthly Charges Distribution

The Histogram shows the Monthly charges distribution in bins and makes it easy to differentiate among the number of customers who are male and female paying most of the amount. From this graph, we come to know that the female customers use more of the internet service than male customers.

```
f, ax = plt.subplots(figsize=(10, 5))
fig = sns.boxplot(x='InternetService', y='MonthlyCharges', data=df, showfliers = False)
```

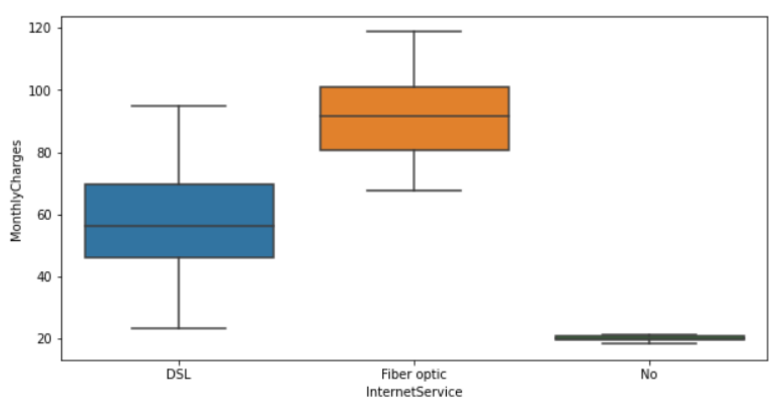

Figure 5: Internet Service Vs Monthly Charges

The above boxplot depicts the distribution of monthly charges for customers who have opted for DSL and Fiber optic service. The monthly charges for fiber optic service is higher than DSL.

	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

Figure 6 : Descriptive Statistics

4. Benchmark and Improvement

4.1 Results from Three Benchmarks without Preprocessing Data

For this classification analysis, we use NaiveBayes,Decision Tree and Random Forest. Models are trained using 10 fold Cross-Validation. We used the Confusion matrix to identify the Accuracy of the model.

4.1.1 NaiveBayes

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

Naive bayes is based on Bayes rule, $P(y|x)=P(y)P(x|y)/P(x)$

$$P(\mathbf{x} \mid y) = \prod_{i=1}^{n} P(x_i \mid y)$$

where x_i is the value of the *i*th attribute in x, and *n* is the number of attributes.

$$P(\mathbf{x}) = \prod_{i=1}^{k} P(c_i) P(\mathbf{x} \mid c_i)$$

where *k* is the number of classes and c_i is the *i*th class

ACCURACY : 71.9296 %

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	5066	71.9296 %
Incorrectly Classified Instances	1977	28.0704 %
Kappa statistic	0.4084	
Mean absolute error	0.2821	
Root mean squared error	0.4842	
Relative absolute error	72.3378 %	
Root relative squared error	109.6587 %	
Total Number of Instances	7043	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.686	0.189	0.909	0.686	0.782	0.441	0.819	0.920	No
	0.811	0.314	0.483	0.811	0.605	0.441	0.819	0.613	Yes
Weighted Avg.	0.719	0.222	0.796	0.719	0.735	0.441	0.819	0.839	

=== Confusion Matrix ===

a	b	<-- classified as
3551	1623	a = No
354	1515	b = Yes

4.1.2 Decision Tree

Decision Tree is a supervised learning algorithm where a tree is built in a series of nested tests. Each node indicates a test performed on one or more attributes to partition the instances into sub groups with as little impurity level as possible. Purity level of a subgroup is measured based on different calculations. Most commonly used measure is the Entropy (Information gain).

Entropy = $\sum_i - p_i \log_2 p_i$ Pi - Proportion of class i

ACCURACY : 77.9923 %

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	5493	77.9923 %
Incorrectly Classified Instances	1550	22.0077 %
Kappa statistic	0.3988	
Mean absolute error	0.2755	
Root mean squared error	0.4128	
Relative absolute error	70.661 %	
Root relative squared error	93.5016 %	
Total Number of Instances	7043	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.885	0.510	0.828	0.885	0.855	0.403	0.758	0.849	No
	0.490	0.115	0.606	0.490	0.541	0.403	0.758	0.516	Yes
Weighted Avg.	0.780	0.406	0.769	0.780	0.772	0.403	0.758	0.761	

=== Confusion Matrix ===

a	b	<-- classified as
4578	596	a = No
954	915	b = Yes

4.1.3 Random Forest

Random forest is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result.One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

Therefore, in a random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. It can also make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds.

Confusion matrix
[[1186 125]
[237 210]]

ACCURACY : 0.79408

4.2 Proportions of the target class variable

Churn	Count
No=0	5178
Yes =1	1869

4.3 Improvement with Pre-Processing :

4.3.1 Replace missing value

We first replaced missing values with mean values in the dataset. It didn't change any of the accuracies by much, probably because there weren't many missing values.

	Process	Naive Bayes(%)	Decision Tree(%)	Random Forest (%)
1	Replace missing value	71.9296	77.9923	79.408

4.3.1 SMOTE

Then we used SMOTE (Synthetic Minority Oversampling Technique), a very common sampling technique that synthesizes new minority instances between existing (real) minority instances to try to solve the class imbalance issue. We obtained high accuracy in the SMOTE method. Decision Tree, Random Forest and Naive Bayes algorithms' accuracy are changed by more than 5 percent. Decision Tree decreased the accuracy unlike Naive Bayes and Random Forest, this may be caused by the SMOTE trimming away some of the conditions. After the process of SMOTE, we overcome the overfitting problem posed by random oversampling. Thus, conditions in the original dataset may be different from the new dataset for Decision Tree.

	Process	Naive Bayes(%)	Decision Tree(%)	Random Forest (%)
2	SMOTE	76.8739	71.6225	88.647

4.3.2 Feature Selection

By looking at the 20 attributes, we drop irrelevant columns such as “customer ID”, and use only the main category such as “phone service” and “ internet service” by eliminating the sub category such as “multiple lines” of phone service and “online security”, “device protection”, “tech support”, “streaming tv”, “streaming movies” of internet service.

Therefore, we use only relevant 11 attributes: gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService, InternetService, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges.

Target Class : Churn

The models improve their accuracy:
NaiveBayes: 71.9296 % -> 77.1972% , increase by 5.2676 %
Decision Tree: 77.9923 % -> 78.83 %, increase by 0.8377%
Random Forest :79.408 -> 78.560%, decrease by 0.848%

	Process	Naive Bayes(%)	Decision Tree(%)	Random Forest (%)
3	Feature selection	77.1972	78.83	78.560

4.3.3 Feature selection with SMOTE

We finally used the Feature selection by selecting the most valuable 11 attributes to attempt to increase the accuracy of the three algorithms without considering the data insight of attribute population ratio. There are Feature selections that we used, the original Feature selection and Feature selection with SMOTE. The Feature selection with discretization overall has higher accuracy than the original Feature selection except for the Random Forest. In terms of average accuracy increment, Feature selection with SMOTE is the preprocess method that increased the three algorithms’ accuracies the most.

In evaluating our preprocess methods, we believe that the SMOTE were particularly valuable in improving the accuracy of the data mining methods. It is because it changes the overall accuracy by more than 5 percent by itself or combining with the Feature selection method.

	Process	Naive Bayes(%)	Decision Tree(%)	RandomForeset (%)
4	Feature selection with SMOTE	77.1972	79.0732	90.9

4.3.4 SUMMARY TABLE

	Process	Naive Bayes(%)	Decision Tree(%)	RandomForeset (%)
0	Benchmark	71.9296	77.9923	79.408
1	Replace missing value	71.9296	77.9923	79.408
2	SMOTE	76.8739	71.6225	88.647
3	Feature selection	77.1972	78.83	78.560
4	Feature selection with SMOTE	77.1972	79.0732	90.9

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        6407               90.9698 %
Incorrectly Classified Instances       636                9.0302 %
Kappa statistic                          0.762
Mean absolute error                      0.1517
Root mean squared error                  0.2669
Relative absolute error                 38.9144 %
Root relative squared error             60.4391 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
                 0.954    0.212    0.926      0.954    0.939      0.763    0.951     0.980
                 0.788    0.046    0.860      0.788    0.822      0.763    0.951     0.886
Weighted Avg.    0.910    0.168    0.908      0.910    0.908      0.763    0.951     0.955

=== Confusion Matrix ===

    a    b   <-- classified as
 4935  239 |    a = No
  397 1472 |    b = Yes
```

Figure 7 : Summary of Random Forest after Feature Selection with SMOTE

5. Takeaways

We used NaiveBayes, Decision Tree and Random Forest for our classification.

While comparing the results after running it on the benchmark model, all three models had similar accuracy levels, but RandomForest had the highest accuracy among them.

Considering the relationship between variables, we mainly use the 11 variables. We only used variables that either have significant meaning in determination of the churn rate or variables with high correlations with the churn rate. It might be explained that by moving redundancy variables, we use the main variables to construct models with high accuracy. Also, considering imbalanced instances in two classes, we did SMOTE to balance the instances to help the model favoring both classes of churn. After re-balancing the instances, we have higher accuracies.

Our best model is the **Random Forest model**. It outperforms the other two models after we pre-process the data. The highest accuracy achieved is up to 90.9 percent.